DOCUMENT RESUME

ED 346 815 IR 014 349

TITLE	Computer Literacy: Intermediate and Secondary Grades, Second Edition.
INSTITUTION	New York City Board of Education, Brooklyn, N.Y.
REPORT NO	ISBN-88315-535-4
PUB DATE	83
NOTE	337p.; For a curriculum guide for the elementary grades, see ED 318 444.
AVAILABLE FROM	New York City Board of Education, Curriculum Unit, Room 310, P.S. 206, Neck Road and East 22nd Street, Brooklyn, NY 11229.
PUB TYPE	Guides - Classroom Use - Teaching Guides (For Teacher) (052) -- Tests/Evaluation Instruments (160)
EDRS PRICE	MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS	*Computer Literacy; Computer Peripherals; *Computer Science Education; Curriculum Guides; Glossaries; Intermediate Grades; *Microcomputers; *Programing; Secondary Education

ABSTRACT

This curriculum guide outlines a course on computer literacy for junior or senior high school students reflecting the technology of the early 1980s. A chart shows computer literacy scope and sequence for students in kindergarten, grades 1-2, grades 3-6, and grades 7-9. Forty lessons for the intermediate grades and high school are then presented, covering topics such as the history of computers, computer components, computer careers, keyboarding skills, programming in BASIC, advantages and disadvantages of computers, computer languages, purchasing a personal computer, word processing, database management, and spreadsheets. Each lesson includes the following sections: performance objectives, vocabulary, motivation, aim, development, application, summary, and exercises. The appendixes include: (1) suggestions for individual student projects; (2) supplemental strategies for teaching programming skills; (3) instructions for a research project on computer careers; (4) a guide to the use of computer peripherals; (5) a guide to computer selection and care; (6) a glossary of computer terms in English and Spanish; (7) a directory of software production companies; (8) a brief history of calculating and computing; and (9) a 61-item bibliography of books and periodicals. A questionnaire for the evaluation of this guide is attached. (MES)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

2R

NEW YORK CITY PUBLIC SCHOOLS
BOARD OF EDUCATION OF THE CITY OF NEW YORK
OFFICE OF CURRICULUM DEVELOPMENT AND SUPPORT • DIVISION OF CURRICULUM & INSTRUCTION

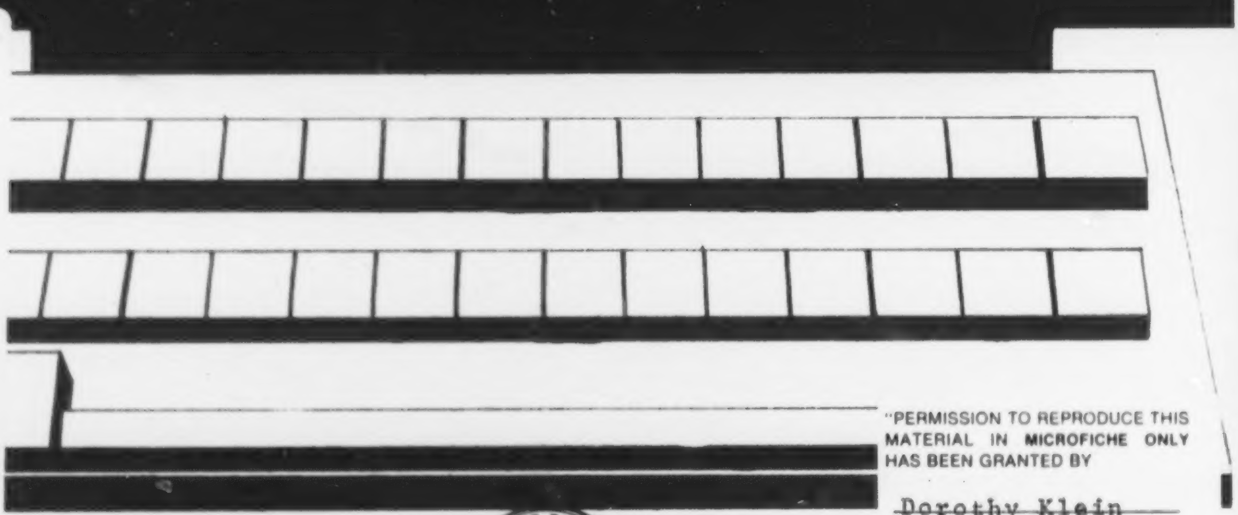
ED346815

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
☒ This document has been reproduced as received from the person or organization originating it
☐ Minor changes have been made to improve reproduction quality
• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy

COMPUTER LITERACY

INTERMEDIATE AND SECONDARY GRADES

SECOND EDITION



"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY

Dorothy Klein

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

IR014349

BEST COPY AVAILABLE

Curric. No. 00-6090-80

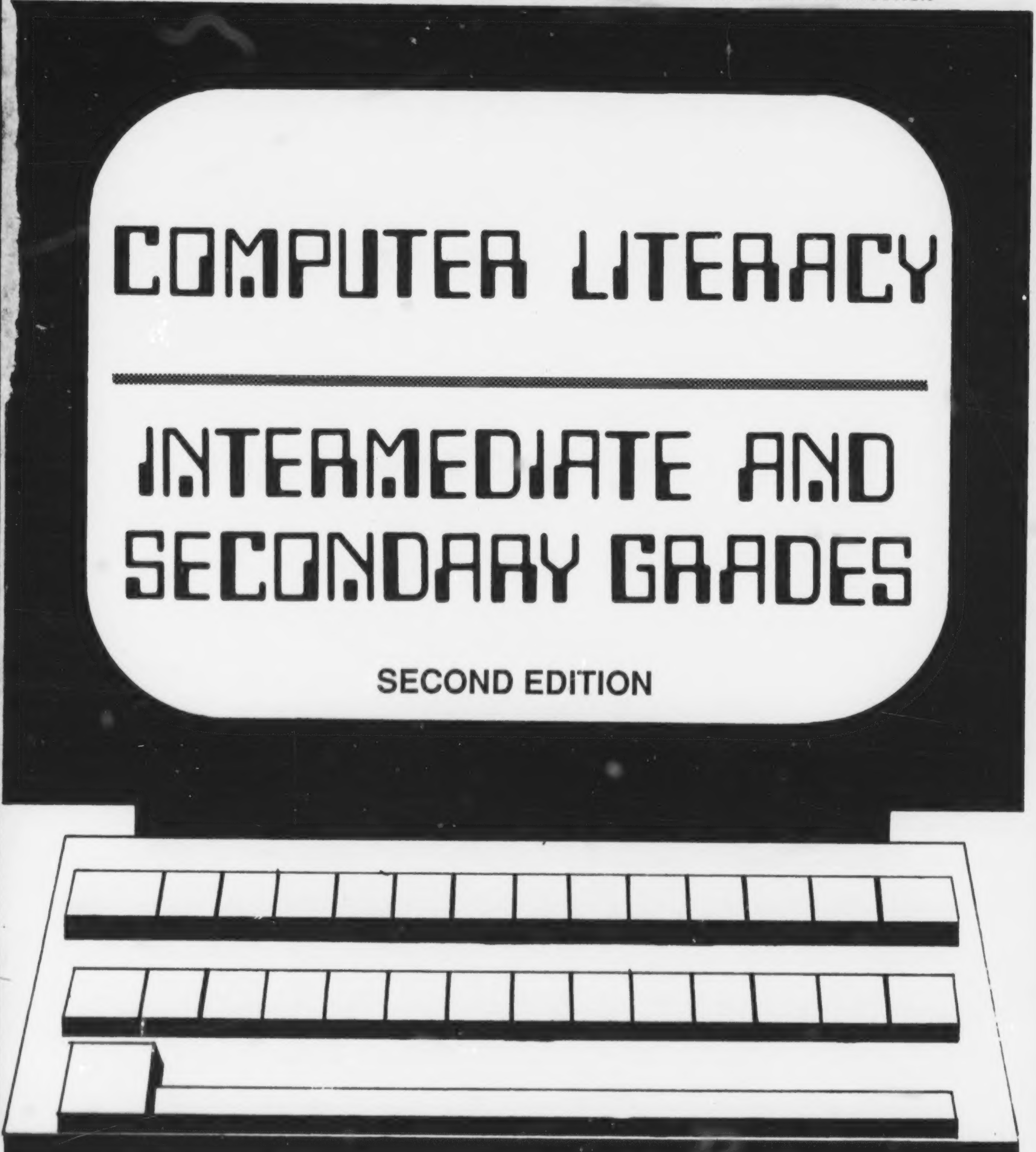
ISBN No. 88315-535-4

NEW YORK CITY PUBLIC SCHOOLS
BOARD OF EDUCATION OF THE CITY OF NEW YORK
OFFICE OF CURRICULUM DEVELOPMENT AND SUPPORT • DIVISION OF CURRICULUM & INSTRUCTION

COMPUTER LITERACY

INTERMEDIATE AND SECONDARY GRADES

SECOND EDITION



Curric. No. 00-6090-80

ISBN No. 88315-535-4

New York City Board of Education

James F. Regan, *President*
Miguel O. Martinez, *Vice President*

Joseph G. Barkan
Stephen R. Franse
Irene Impellizzeri
Marjorie A. Lewis
Robert F. Wagner, Jr.

Nathan Quinones, *Chancellor*

Charles I. Schonhaut, *Deputy Chancellor*

Jerald Posman, *Deputy Chancellor for Financial Affairs*

Louise Latty, *Chief Executive for Instruction*

Charlotte Frank, *Executive Director*
Division of Curriculum and Instruction

Irwin Kaufman, *Director*
Office of Technology

Copyright ©1983 by the Board of Education of the City of New York

Application for permission to reprint any section of this material should be made to the Superintendent of Schools, 110 Livingston Street, Brooklyn, N.Y. 11201. Reprint of any section of this material shall carry the line, "Reprinted from *Computer Literacy, Intermediate and Secondary Grades* by permission of the Board of Education of the City of New York."

Foreword

The phenomenon of computers in the classroom is not new to the New York City public schools. What is new, however, is their prevalence, their impact, and the ways in which their power is being harnessed.

Beginning with a few sites during the 1960s, computers were used and viewed as instructional delivery systems. With the technological advances of the 1970s, computers became smaller, faster, less esoteric, and less expensive. Educators became aware of the untapped potential of computers in education and began to view computers as a powerful tool for learning. Working on their own, several pioneering teachers began to use computers with their classes as information processors and as problem-solving tools. The excitement and the presence of computers spread quickly. We are now proudly viewing a school system which is meeting the challenges of using computers as effective tools for learning for students from pre-kindergarten through senior high school.

Our responsibility to the students and teachers of New York is clear. Students need to have basic computer concepts and skills. Teachers need to have the support of a developmental, sequential set of goals, objectives and activities in computer literacy. The Computer Literacy: Intermediate and Secondary Grades curriculum is our response to this challenge.

This document is based upon current research findings and the practical experience of teachers who pioneered computer literacy in the schools. It is structured to permit a flexible and pragmatic approach to the study of computer awareness and programming skills. Each module within this course of study outlines specific performance objectives and provides a variety of experiences and activities to help the students meet those objectives.

It is hoped that this guide will help teachers meet the most pressing educational challenge of the 1980s--preparing all students to function effectively within the Information Age.

Charlotte Frank

Charlotte Frank
Executive Director
Division of Curriculum and Instruction

New York Public Schools should order additional copies of this publication from the Bureau of Supplies.

ISBN No. 88315-535-4 Curric. No. 00-6090-80

Other persons or institutions may obtain copies of this publication from the Curriculum Production Unit of the Office of Curriculum Development and Support, Room 613, 131 Livingston Street, Brooklyn, N.Y. 11201. For information, call (718) 596-4902.

It is the policy of the New York City Board of Education not to discriminate on the basis of race, creed, national origin, age, handicapping condition, or sex in its educational programs, activities, and employment policies, as required by law. Inquiries regarding compliance with appropriate laws may be directed to Mercedes Nesfield, Director, Office of Equal Opportunity, 110 Livingston Street, Brooklyn, N.Y. 11201; or to Charles Tejeda, Director, Office of Civil Rights, Department of Education, 26 Federal Plaza, Room 33-130, New York, New York 10278

Introduction

The use of computers in the classroom has become more prevalent during the past few years. Through this use we have developed a knowledge of their functions and a better vision of their potential. We encourage our colleagues to freely experiment and attempt new approaches. Only with all of us working in concert, can we fulfill our vision for an effective integration of school technology into all subject areas. From these beginnings, we will nurture the critical thinking and practical skills our students will need to succeed in the information age. If children are not comfortable with computers, they will be as unprepared for the future as an adult who cannot read or write.

The purpose of this guide is to give teachers and supervisors a working knowledge of various approaches to enhance pupil learning. Our guides are intended to be open ended and flexible in nature. As a 1920 car is not efficient by 1980 standards, outmoded computer education ideas of the seventies have been replaced by the new visions of the eighties. Classroom use of technology has doubled and redoubled in the past few years. Teachers are continually devising new ideas for the use of this technology.

Our primary responsibility in this area and our reason for writing this book, is to see that computers are used as an efficient classroom tool. Only through a real sharing can we move forward in this rapidly changing field of technology. Feel free to write to me directly with your comments and suggestions concerning this curriculum guide, so that we may improve and update it to meet your needs.

Irwin Kaufman

Irwin Kaufman
Director
Office of Technology

Acknowledgments

The Computer Literacy: Intermediate and Secondary Grades curriculum guide is a joint project of the Division of Curriculum and Instruction, Charlotte Frank, Executive Director, and the Office of Technology, Irwin Kaufman, Director.

Morris Freedman, Director of the Office of Curriculum Development and Support, and Irwin Kaufman, Director of the Office of Technology, provided overall supervision of this project.

Michael Ryan, former Director of the Computer and Information Science Unit, and Joseph Eaione, Assistant Director, Computer and Information Science Unit, provided coordination.

Stephen Radin, Director of Curriculum Development, of the Office of Technology, served as an author and coordinated the development of the second edition of this guide.

This document was a labor of love for some highly dedicated professionals. Those responsible for planning, designing, researching, and developing the guide include: Linda Broudy, South Shore HS; Sidney Broudy, South Shore HS; Elizabeth Burton, CSD #9; Frank Campisi, JHS 259K; Gloria Catanzaro, PS 269K; Joseph Eaione, Computer and Information Science Unit; Pierre Fabinski, Assistant Principal, Christopher Columbus HS; Patrick Ianniello, PS 177K; Maryann Marrapodi, PS 153M; Jerry Ostrover, formerly of the Computer and Information Science Unit; Randy Ross, Assistant Principal, Mathematics, Seward Park HS; Michael Ryan, Assistant Principal, Computer and Information Science Unit; Robert Slotkin, Special Education; Russell Zausmer, PS 16R.

The following people were on the original steering committee which framed the outline and developed the first drafts: Nick Androu, CSD #27; Helene Fink, PS 91; Ken Hess, PS 203K; Coran James, former of I.S. 320K; Carolyn Pinchback, CSD #19; Stephen Radin, former of CSD #22; Steve Wulfson, Assistant Principal, I.S. 174X.

Special thanks are due: John Bailey, Principal, PS 306K; Bobbie Beinhacker, Assistant Principal, Jamaica HS; Al Bartolo, CSD #31; Phil Bayman, Christopher Columbus HS; Alvaro Cunqueiro, Morris HS; Gerald Elgarten, Assistant Principal, Francis Lewis HS; Ed Gibson, formerly of Office of Data Processing; Benjamin Harvey; Jessie Gordon; George Milton, Principal, JHS 17M; Stan Montz, Assistant Principal, Murray Bergtraum HS; Bernard Newman, Benjamin Cardozo HS; Melvin Serisky, formerly Principal, Francis Lewis HS; Sigmund Tobias, Center for Advanced Study in Education; Steve Wills, PS 76Q.

The bilingual glossary was prepared by Aurea Rodriguez, Director of the Curriculum Development Unit; Tomasita Hidalgo, Curriculum Specialist; and Bessy Solis, Secretary; all of the Office of Bilingual Education.

The following members of the Committee for Cooperative Curriculum Development in Career Education gave much needed assistance and critical review: George Quarles, Chief Administrator, Office of Occupational and Career Education; Patricia Seabrook-Black, Principal, High School Division; Jack Dixon, Office of Occupational and Career Education; Robert A. Folchi, Superintendent, High School Division; Howard Friedman, Deputy Administrator, Office of Occupational and Career Education; Renee Sherline, High School Division; Norman Watnick, Assistant Director, Business and Distribution Education, Office of Occupational and Career Education.

The companion tests were prepared with the help of: Ida Echevarria, Director, Office of Testing, Division of Curriculum and Instruction; William Weichun, Office of Testing, Division of Curriculum and Instruction; Richard Rosenblum, I.S. 61K; Juliet Simms, Morris HS

Revision and editing by: Marianne Sabados Benedikt, Early Childhood Teacher; Geri Ellner, William H. Maxwell VHS; David Penner, JHS 125Q; Stan Solomon, Manhattan Center for Science and Mathematics.

Steven Goodman supervised the word processing and, with Judy Goldberg, and Janet Taylor, coordinated the production.

Gwen Solomon was responsible for writing, coordinating, editing and sequencing the final version of this guide.

The manuscript was prepared for publication by the Editorial and Production Unit, Alaire Mitchell, Unit Head. Anne Wolf revised the history lessons. The cover was designed by John Gampert, and Ernest Mortuzans.

Notes on the Use of This Guide

This publication was designed to serve the purpose of aiding teachers in instructing their classes in computer literacy. No one document can meet the needs of all teachers or all classes; however, this curriculum guide on computer literacy was intended to be adaptable.

In the senior high schools, this guide can be used as an intensive (one semester) course in computer literacy for students who have had little or no prior exposure to computers. It may also be expanded and used over a period of an entire school year.

In the junior high and intermediate schools, there are several options. This course in computer literacy may be taught as a separate class over whatever time frame seems appropriate, or it may be integrated into a regular school curriculum to be presented over the course of two or three years.

Since each situation is different, it is the judgment of the staff of each school that will determine how this guide can be used most effectively. In fact, the individual lessons may be used in their current format or may be adapted to meet the particular needs of various groups of students.

Scope and Sequence

TOPIC	PERFORMANCE OBJECTIVES: Students will be able to:	K	1-2	3-6	7-9
1. The Impact of Computers on our lives	. list common uses of computers.	X	X	X	X
	. explain how computers have affected people's lives.			X	X
	. identify computer-related occupations.			X	X
	. identify computer skills.			X	X
	. explain the use of computers in several areas in society.			X	X
	. discuss the advantages of computers in society.			X	X
	. discuss the disadvantages of computers in society.			X	X
	. discuss the future in terms of computer use in society.			X	X
	. understand the history of computers and computing.			X	X
	. explain how to be an aware consumer of computers.			X	X
2. Computer Components	. identify parts of a computer.	X	X	X	X
	. use a computer.	X	X	X	X
	. use computer terms.	X	X	X	X
	. explain how a computer works (input, process, output).			X	X
	. list several computer languages and their uses.				X
	. recognize differences among various models of computers.				X
	. recognize differences among various types of computers.				X
3. Keyboard Skills	. identify and use letters, numbers and characters on a keyboard.	X	X	X	X
	. identify and use special function keys.		X	X	X
	. develop facility in using a computer keyboard.	X	X	X	X
4. Special Applications	. discuss the use of computers in various academic fields.			X	X
	. discuss the use of computers in various business fields.			X	X
	. use a word processing program.			X	X
	. use a database management program.				X
	. use an electronic spreadsheet program.				X
5. Computer Care	. demonstrate proper care of hardware and software.	X	X	X	X
	. demonstrate how to turn the computer on and off.	X	X	X	X
	. demonstrate how to load a program.	X	X	X	X
	. demonstrate how to save a program.		X	X	X
	. demonstrate how to operate peripherals.	X	X	X	X

6. Programming Skills	. explain the concept of programming.			X	X
	. operate computers and peripherals.	X	X	X	X
	. use an operating system for programming.			X	X
	. use system commands.			X	X
	. develop a course of action prior to writing a program.		X	X	X
	. use problem-solving strategies in a program.		X	X	X
	. use flowcharting to solve a problem.			X	X
	. use a computer as a calculator.			X	X
	. explain and use the order of arithmetic operations.			X	X
	. explain and correct simple error messages.		X	X	X
	. use editing procedures to correct programs.		X	X	X
7. Logo	. understand and use directionality.	X	X	X	
	. create designs.	X	X	X	
	. explain the concepts of size, shape, angle, and spatial relationships.	X	X	X	
	. draw specific shapes.	X	X	X	
	. write procedures.		X	X	
	. use single and multiple inputs.		X	X	
	. use recursion.		X	X	
	. list the properties of Sprites.		X	X	
	. use Sprites in procedures.		X	X	
	. explain and use tiles and characters.		X	X	
	. use the list processing capability to write text.		X	X	
8. BASIC	. use programming statements.			X	X
	. PRINT strings and numbers.			X	X
	. assign variable names to strings and numbers.			X	X
	. use remarks as a programming technique.			X	X
	. format screen output.			X	X
	. use INPUT and READ/DATA to include information in a program.			X	X
	. use conditional and unconditional branching.			X	X
	. use loops in a program.			X	X
	. use integer and random number functions.			X	X
	. create graphic displays on the computer.			X	X

Table of Contents

Page

Foreword iii
Acknowledgments v
Introduction vii
Notes on the Use of this Guide ix
Scope and Sequence x

LESSONS

1 Introductory Lesson 1
2A History of Computing 5
2B History of Computers 10
3 Computer Components 13
4A Computer Careers I: Creating Career Clusters 19
4B Computer Careers II: Qualifications and Training 25
4C Computer Careers III: A Look at the Future 31
5A Keyboard Skills: TRS-80 37
5B Keyboard Skills: PET 45
5C Keyboard Skills: Apple II Plus 52
6 The PRINT Statement: Messages 58
7 The PRINT Statement: Numerical Expressions 62
8 Programming Mode 66
9 Program Modification 69
10 Numeric Variables and Assignment Statements 74
11 String Variables and Assignment Statements 80
12 Unconditional Branching (GOTO)
Print Format -- Comma and Semicolon 84
13 Introduction to Flowcharting 89
14 Order of Operations in Direct (Immediate) Mode 96
15 Order of Operations in Programming Mode 100
16 The REM Statement 104
17 INPUT Statement with Numeric and/or String Variables 108
18 INPUT Statements Using Prompts 113
19 READ/DATA Statements (One Variable) 118
20 READ/DATA Statements (Multiple Variables) 124
21 Flowcharts and Problem Solving with Conditionals 130
22 Programs with Conditional Branching (IF/THEN Statement) 134
23 Writing Programs with IF/THEN Statements and Numerical Variables 138
24 Program Trace with Unconditional Branching (GOTO) 143
25 Program Trace with Conditional Branching (IF/THEN) 147
26 Comparison of String Variables 151
27 Conditional Statements Involving String Variables 154

LESSONS **Page**

28 Counting with IF/THEN 157
29 Introduction to the FOR/NEXT Loop 161
30 Counting, Totalling, and Inputting Using a FOR/NEXT Loop 167
31 Integer and Random Numbers in Programs 173
32 Generating Random Numbers in Programs 178
33 The Advantages of Computers 181
34 The Disadvantages of Computers 187
35 History of Computer Language 192
36 Computer Languages/Binary Numbers 196
37 Consumerism -- Purchasing a Personal Computer 200
38 Word Processing 206
39 Data Base Management 210
40 Electronic Spreadsheet 213

APPENDICES

A Individual Student Projects 219
B Supplemental Strategies for Reinforcement 224
C Instructions for Computer Career Research Project and Metroguide Printout 246
D Use of Peripheral Devices 254
E Computer Selection and Care 262
F Glossary of Computing Terms 268
G Spanish Glossary of Computing Terms (Glosario de terminos usados en computadoras) 280
H Directory of Selected Software Production Companies ... 296
J A Brief History of Calculating and Computing 303
K Bibliography 310

Questionnaire 317

Lessons

Lesson 1

INTRODUCTORY LESSON

PERFORMANCE OBJECTIVES: Students will be able to:

- explain the various ways in which the computer and computer technology has affected and will continue to affect our lives.
- suggest probable effects of computer technology upon the job market.
- cite how this course will enable the participants to become part of this revolutionary trend.
- state the goals and objectives of this course.

VOCABULARY: NEW: technology, technological revolution, job market

MOTIVATION:

- What do all of the following share in common?
  - the wheel
  - steel
  - the printing press
  - electronic communication (radio, tv, etc.)
  - self-propelled vehicles (auto, etc.)
  - manned flight (aircraft, spacecraft)
  - the computer

  (Each is an example of how humans have used technology to permanently change the way we live. The dictionary defines technology as "the branch of knowledge that deals with the industrial arts." Elicit or provide examples of how the wheel, printing press, aircraft have changed the world.)

- Each of the above caused tremendous and permanent changes. Of the list above, which is the one technological development that is still having a tremendous impact today

17

and probably into the future? (Development of the computer. The computer industry is still growing, changing, advancing.)

- How could we demonstrate that computers have affected you? (Simple! You are taking this course!)
- Since this lesson is an introduction to the course, what might its purpose be? (Elicit AIM.)

AIM: How might this course help us to understand and to become part of the computer revolution?

DEVELOPMENT:

1. Arrange class into groups or committees. Explain that they will be allowed a specific amount of time to derive answers to the following issue:

   > "The computer, and computerized equipment, have caused great changes in our world. Write down all of the changes brought about by the development and use of computers and computerized equipment."

2. While committees are working, arrange for volunteers who will record responses on the chalkboard and for an official secretary who will maintain a permanent record of the chalkboard entries.

3. After an appropriate time, announce that committees will present their findings. Allow for open but orderly discussion of the results. An acceptable response would include:

   - computerized fuel systems on autos
   - computerized control of space shuttle
   - computerized diagnostic medical equipment
   - computerized assembly-line robots
   - word processing equipment
   - digital clocks, calculators, microwave ovens
   - computerized school records
   - computerized billing systems for credit card
   - video games
   - personal computers in households

(Teacher may have to elicit some of the above using questions.)

4. Let's continue our brainstorming session. By the year 2000, how else might we see the computer used? (Encourage class to indulge in flights of imagination and creativity as there are no wrong answers.)

   - Some may argue that computers will reduce job openings and cause lay-offs. Accept this as a valid comment, but attempt to have the class understand that, as certain older employment areas decline, newer areas will open and grow.

6. The computer will provide job opportunities for people who have the proper skills. What sort of jobs are there even today in the computer field? (Assembling computers, programming computers, repairing computers, operating computers, etc.)

   - These topics will be covered in greater detail in future lessons.

7. How can a young person today acquire the skills that will be needed in the computer field tomorrow? (By taking courses while still in high school, using school library, and consulting guidance personnel to plan for future careers.)

8. How might this specific course help us as we prepare for careers in the computer revolution? What can we learn from this course?

   - We can learn how computers work.
   - We can learn how to operate the computer.
   - We can learn how computers are programmed.
   - We can learn about the careers available in computers.
   - We can learn to become part of the computer revolution.

APPLICATION:

- Since this is an introductory lesson, it would be nice to introduce you to the computer! Your teacher will explain exactly what to do. (The following program should be entered prior to this point of the lesson. Explain to students how to cause program to execute.)

```
10 (Clear/Screen command)
20 PRINT "Welcome to this course!"
30 PRINT
40 PRINT "I am (type of computer)"
50 PRINT "Type Your Name Now"
60 INPUT N$
70 (Clear/Screen command)
80 PRINT " Hi "; N$;
90 GOTO 80
```

- (Note: Line 80-space before and after Hi, semicolons as shown.)

SUMMARY:

- What are some of the skills which we will learn in this course?
- Why will such skills be important in tomorrow's job market?

EXERCISES (ASSIGNMENTS):

- Answer each of the following in complete sentences. You will not be graded on your responses. Rather, this is to assist your teacher in designing and planning future activities:
  - How much experience and/or exposure have you had to computers?
  - If you had the opportunity to work with computers, what sort of job would you want to do? (Describe activities performed.)
  - Do you know anyone (family, friends, etc.) who works with a computer or in the computer field? If yes, describe what this person does.
  - Do you know anyone who could be described as a "computer-freak," i.e., a person who is totally obsessed with computers? If so, tell about this person in a few sentences. How do other people act or react to this person?
  - Do you know anyone who is either fearful of computers or who claims to "hate" computers? If so, describe this person's feelings and/or give examples of this person's attitude.

Lesson 2A

HISTORY OF COMPUTING

PERFORMANCE OBJECTIVES: Students will be able to:

- label innovations in computing.
- sequence innovations in computing.
- identify key features of a selected innovation.

VOCABULARY: NEW: abacus, Napier bones, arithmetic machine, Jacquard loom, analytical engine, tabulating machine, Charles Babbage, Lady Ada Lovelace, Herman Hollerith.

MOTIVATION:

- Imagine that you need to solve this problem: You have one dog and she has two babies. How many dogs are there?
  - List the devices you could use to help you calculate the answer. (fingers and toes, paper and pencil, adding machine, calculator, etc.)
- To solve this problem, what devices could you use? Census forms for Americans all around the country have been returned. You are responsible for computing the returns to see how people live in America, where they live, and how they earn their livings.
- Why does the list of devices for the second problem differ from the list for the first problem?
- When people have simple computing problems, simple devices can work. As computing problems become more complex, people invent more sophisticated devices.

AIM: How have computing devices become more complex?

DEVELOPMENT:

1. Read the attached excerpts of articles from THE COMPUTER TIMES.
2. What was the earliest device people used to help them calculate? What were the advantages of this device? What were the disadvantages?
3. How did the abacus improve on fingers and toes as a calculating device?
4. In the 1600s inventors developed new devices for calculating. What calculations were these devices able to do?
5. What contributions did inventors in the 1800s make to the development of calculating devices?
6. What features of these devices have been "borrowed" for use in modern computers?
7. Why have people spent time and money building new and more complex devices for calculating?
8. How would our lives be different today without the invention of sophisticated computing machines?

APPLICATION:

- Match each description in column III below with the device it describes. Write the letter of the device on the appropriate line in column I.

  a. Jacquard loom
  b. tabulating machine
  c. analytical engine
  d. arithmetic machine
  e. Napier bones
  f. abacus

- Put the innovations in chronological (time) order. In column II write a 1 next to the earliest device and so on.

I	II	III (Description)
______	______	Machines with punched cards that control the threads being woven into cloth.
______	______	Wood rods that can be used to multiply large numbers.
______	______	First machine that stores, processes, and prints the results of computations.
______	______	Box with wires and beads used in calculating.
______	______	Machine that reduced the time needed to tabulate census results.
______	______	Device the size of a shoe box. Has a series of wheels and gears that add and subtract.

SUMMARY:

- How have computing devices changed?
- Why have computing devices changed?

EXERCISES (ASSIGNMENT):

- Select one device described in THE COMPUTER TIMES. Imagine you were alive at the time the device was first developed. Write an advertisement to let people know about the device and to tell people its advantages. Be sure to:
  - describe the device.
  - explain how the device will help people calculate more easily.
  - list the advantages of the device.
  - name the person or persons who developed the device (only if appropriate).

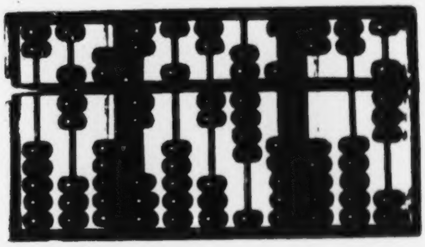
The Computer Times

Earliest Times

A recent survey by this newspaper reveals that people are using their fingers and toes to count. Our reporter has learned that this works well as long as the numbers are small enough. For large numbers people are using sticks, pebbles, or scratches on the walls to show how many things they have.

Thousands of Years Ago

The abacus, shown below, has been invented in the Near East and China. The abacus is made of beads that slide on wires, and counting is done by pushing the beads around. The abacus itself does not do any computing, but our sources report that the device does help people keep track of numbers when they do calculations. In Japan a similar device is called the soroban.

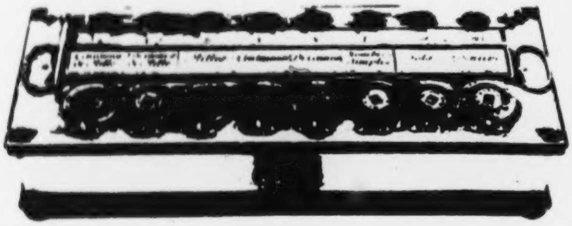


The 1600s

Our reporter has learned that John Napier of Scotland has invented a device that multiplies large numbers. Called Napier's bones, it consists of wooden rods which can be moved around. The rods are read to get the product of two large numbers.

1642

We have just learned that Blaise Pascal, age 19, has developed an arithmetic machine. His new invention, shown below, is about the size of a shoe box. It adds or subtracts numbers through a clever use of wheels and gears. The machine displays the correct answers.



1804

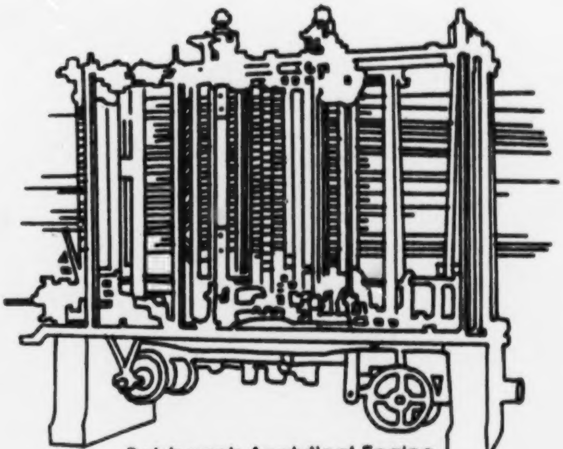
We have learned that Joseph Jacquard has improved on the machine which is used for weaving cloth. He has invented a loom that uses punched cards to control which threads are woven into the cloth. The punched cards keep certain threads from going into the loom and let in others when they belong in the pattern. These punched cards store the information for working the loom and are very much like a program.

1835

In England Charles Babbage, working with Lady Ada, the Countess of Lovelace, has invented the analytical engine. This new device can receive instructions, process and store information, and print results. This is all done mechanically by the machine. (See the picture below). This is the first computing machine that can be programmed. Lady Ada helped develop the instructions for the machine.

1890s

Herman Hollerith has invented a tabulating machine. Hollerith developed this invention for the U.S. Census Bureau which wants to speed up the time it takes to tabulate the results of the census. It has taken the Census Bureau as long as seven years to accomplish this task. It is estimated that Hollerith's machine may help the Bureau complete its work in as little as three years.



Babbage's Analytical Engine

Lesson 2B

HISTORY OF COMPUTERS

PERFORMANCE OBJECTIVES: Students will be able to:

- identify the important characteristics of each generation of computers.
- discuss trends in computer development.

VOCABULARY: NEW: digital computer, generation, ENIAC, vacuum tube, UNIVAC, transistor, integrated circuit, integrated circuit chip.

MOTIVATION:

- Write the title, "Four Generations of Modern Computers" on the chalkboard. Tell students they will be reading an article with that title. Ask them to guess what the article will be about. List all guesses on the chalkboard.
- Distribute article. Have students skim the article to see which of their guesses were correct and which were not. Erase any incorrect guesses and add new items to the list.

AIM: How have computers changed since the first computer, Mark I, was built in 1944?

DEVELOPMENT:

1. Read the article and complete the activities.
2. How do today's computers differ from earlier computers?
3. Based on the changes which have occured in the past, what advances do you think there may be in the computers of the future?

4. Many people today own home computers. Why were there no home computers until the fourth generation of computers had been developed? What advances might take place in computers for them to become as common in people's homes as televisions are today?

5. Why do you think the ability to store a program was a very important advance in computer technology?

6. What advances in technology made arcade and television computer games possible?

APPLICATION:

- Match each statement in the first column with the appropriate computer generation: first, second, third, or fourth. Use the article to help you decide.

Statement	Generation
1. During this generation computers became small enough and cheap enough for many businesses to buy them.	
2. During this generation digital computers became the most common kind of computer in use.	
3. During this generation integrated circuits were introduced and smaller computers could be built.	
4. During this generation huge computers were built that had vacuum tubes and could do as many as 1,000 calculations per second.	
5. During this generation integrated circuit chips were introduced and computers became smaller, faster, and cheaper than ever before.	

Statement	Generation
6. During this generation the first computers were built that had almost no moving parts.	
7. During this generation transistors replaced vacuum tubes. Computers no longer needed special airconditioning and they were now more dependable.	

SUMMARY:

- How have computers changed?
- What is the next step?

EXERCISES:

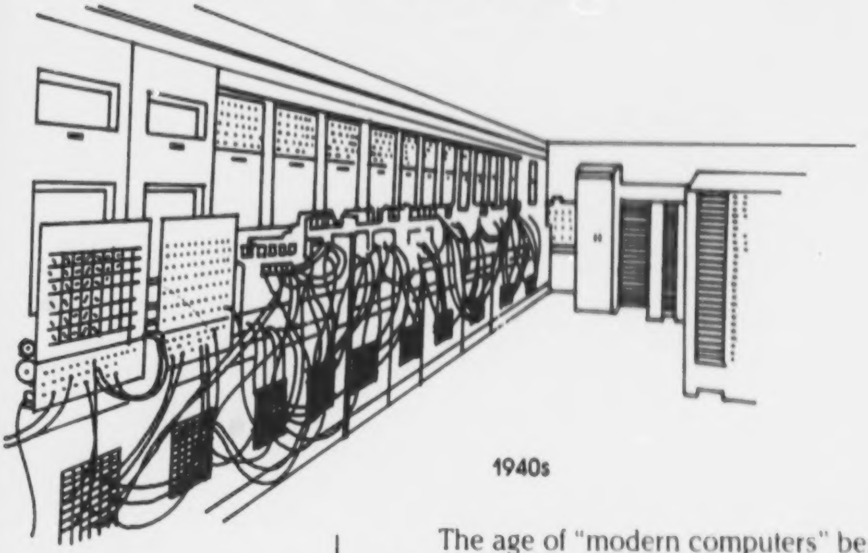
- Read each statement below. Based on the information in the article you read, decide which statements you can defend. Put a ✓ if you think a statement can be defended on the evidence in the article. Put an X if you think a statement cannot be defended.

Statement	✓ or X?	Reasons
Changes in technology have made today's computers more efficient than any computers that came before.		
To do increasingly more complicated calculations, engineers have needed to build bigger and bigger computers.		
Today's computers can do calculations more cheaply than earlier computers.		
Without integrated circuits and chips, there would probably be no arcade or television computer games.		
The fourth generation of computers will be the last.		

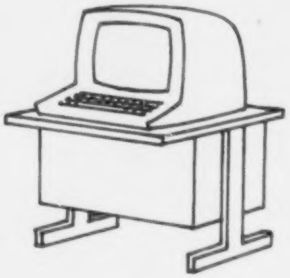
FOUR GENERATIONS OF MODERN COMPUTERS

As people learn new things, they use that knowledge to generate more new information. The more information they have, the more they try to find better ways to store it, process it, and retrieve it. During the past forty years, a giant leap has been made in dealing with information. Men and women have developed high-speed computers which accept, store, process, and give out information. The computers work faster than people like Pascal or Hollerith ever dreamed possible.

To **retrieve** (ri trēv') ⇨ information means "to get information that is stored somewhere".



1940s



1980s

The age of "modern computers" began in 1944. In that year an American engineer at Harvard University, **Howard Aiken,** built a computer. It worked very much like a machine designed more than 100 years earlier—Babbage's Analytical Engine. Aiken's computer, called the Mark I, accepted information through punched cards. It stored and processed the information. It printed the results on an electric typewriter. The Mark I was able to do many different tasks. It was a huge machine. It took up the space of a school gymnasium. It took only a few seconds to calculate a math problem—quite a feat for 1944! The Mark I is known today as the world's first **electromechanical computer**.

Remember, Babbage ⇨ was not able to complete his Analytical Engine.

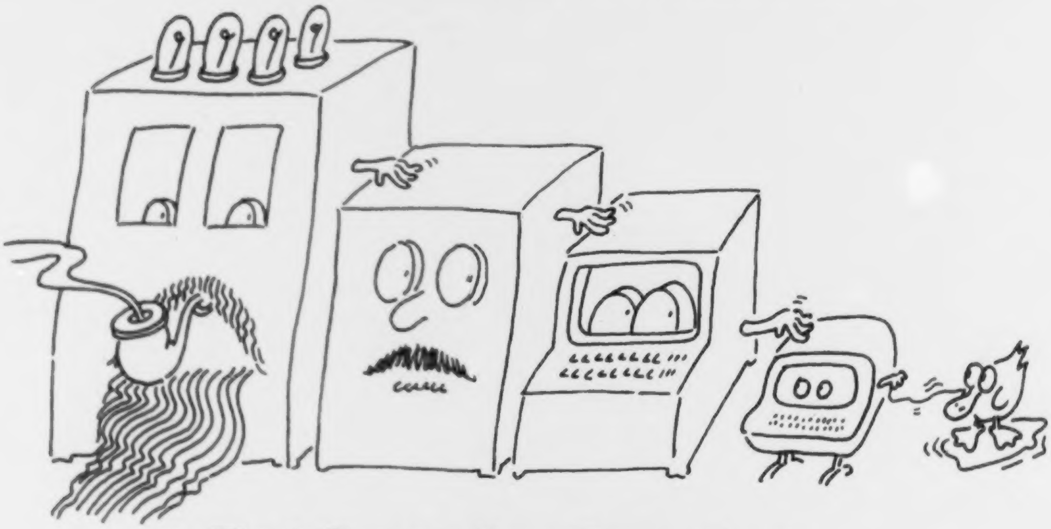
An **electromechanical computer** ⇨ has both electrical and mechanical (moving) parts.

Soon after the invention of the Mark I, scientists began to build computers that had almost no moving parts. That is, they were electronic rather than mechanical. Most of the computers that you'll be reading about are called **digital computers**. A digital computer changes information into digits to be stored and processed.

Digital (dij'it əl) means ⇨ "having digits, or numbers."

Electronic digital computers quickly replaced the Mark I. In fact, a few years after the Mark I was built, electromechanical computers became old-fashioned, and weren't used any more.

There have been several major changes in digital computers during the past forty years. Each change ushered in a new "generation" of computers. Just as we have different generations of people in a family, computers have generations, too.



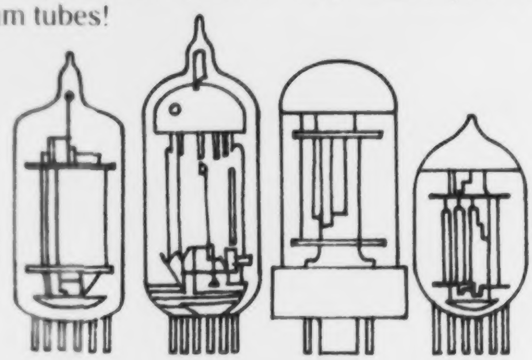
First-Generation Computers

The only "movement" in a circuit is the flow of electricity.

ENIAC stands for Electronic Numerical Integrator and Calculator.

As moving parts inside computers were replaced by electrical circuits, computers worked faster and more efficiently. The first all-digital computer was completed in 1946 at the University of Pennsylvania under the direction of two engineers, **John W. Mauchly** and **J. Presper Eckert.** The computer, called the **ENIAC,** was even bigger than the Mark I. It weighed over 30 tons! It conducted electricity through vacuum tubes. In fact, the computer used over 18,000 vacuum tubes!

Radios and televisions used vacuum tubes through the 1950s.



Vacuum Tubes

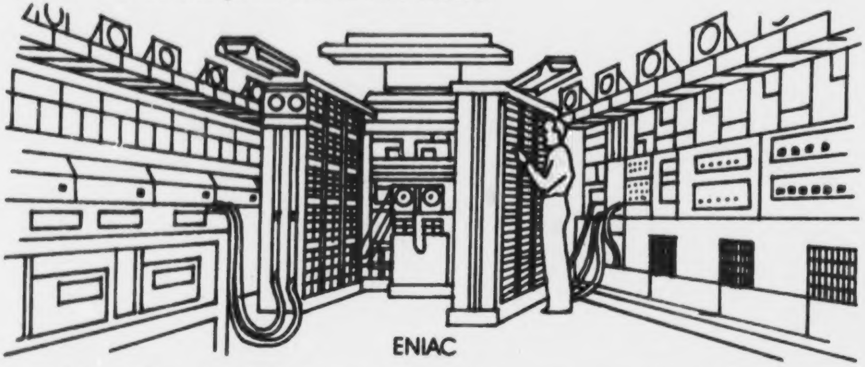
The ENIAC's electric bill was $1,800 a month in 1946!

Vacuum tubes get hot, and 18,000 of them created a lot of heat. So, it was necessary for the ENIAC to have special air conditioning units to keep it cooled down.

Does a computer really have a brain?

Today's computers could solve the problem in a few minutes, or less.

The ENIAC was considered quite a "brain." It was 300 times faster than the Mark I. It worked a thousand times faster than a person using a desk calculator. It was given a problem that would have taken 100 engineers, working eight hours a day, an entire year to solve. The ENIAC solved the problem in two hours.



ENIAC

Soon after the ENIAC was built, **John von Neumann** had the idea of storing a computer program in the computer's memory. Up until this time, only the numbers used in the program were stored in the memory. Von Neumann's idea enabled people to build computers that worked faster than the ENIAC. In fact, today's computers are based on von Neumann's idea of storing programs in the memory.

UNIVAC stands for Universal Automatic Computer.

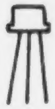
A **commercial** product is one that can be bought and sold.

A few years later, in 1951, Eckert and Mauchly designed another computer called the **UNIVAC.** The UNIVAC was even larger than the ENIAC. Eckert and Mauchly sold the UNIVAC to the United States Census Bureau. Other models of the UNIVAC were built and sold, making the UNIVAC the first **commercial** computer. UNIVAC computers are not used today, but the original one can be seen at the Smithsonian Institute in Washington, D.C.

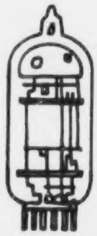
Second-Generation Computers

One transistor could do the work of about 40 vacuum tubes. It used only about 1/100 of the power. Electric bills for second-generation computers were much lower!

In the late 1950s, the **transistor** replaced vacuum tubes in the computer. A transistor could conduct electricity more quickly and efficiently than a vacuum tube. It was more reliable, too. Vacuum tubes often "burned out" and needed to be replaced. Transistors rarely needed to be replaced. A transistor was also much smaller than a vacuum tube, and it did not get hot.



Transistor



Vacuum Tube

Other improvements included a larger memory and improved input/output devices.

When companies started building computers with transistors instead of vacuum tubes, the computers became smaller. They also solved problems ten times faster than first-generation computers.

It is difficult to give credit to any one person for building a second-generation computer. These computers were so complex that it took many people with many different skills to design all the computer parts. Several large companies built these second-generation computers. Some were for their own use. Some were sold to other companies.

Third-Generation Computers

Remember, a circuit is a path through which electricity flows. The path is usually made by a thin wire. "Integrated" circuits are circuits which are connected to each other.

In 1964, tiny **integrated circuits** were developed to take the place of transistors. These tiny circuits were even faster and more reliable than transistors. Integrated circuits, or IC's, were very small. They took up very little space.

First- and second-generation computers were large machines. They stood on the floor and occupied a lot of space. But the third-generation computers were much smaller. Some could even sit on top of a table. Also, they could operate 100 times faster than second-generation computers (1000 times faster than first-generation computers!)

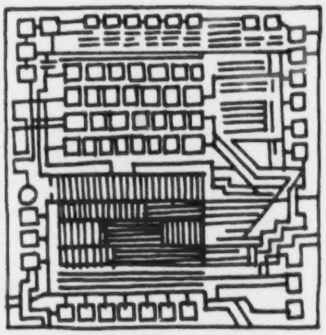
Mass-produced means "manufactured in very large quantities."

Integrated circuits were mass-produced at a low cost. So, as more computers were built with integrated circuits, the price of computers dropped lower and lower. Third-generation computers were inexpensive enough and small enough to be bought by thousands of companies around the world.

Fourth-Generation Computers

In the mid-1970s, scientists developed a method of putting thousands of integrated circuits on one tiny silicon chip. The chip itself is so small that it can fit through the eye of a needle. It is difficult to imagine so many circuits on such a tiny surface. It takes many delicate instruments and special scientific techniques to create this "miracle" chip, called an **integrated circuit chip.**

Actual-Size Chip



Enlarged Chip

Integrated circuit chips led to the development of microcomputers. ⇨

Integrated circuit chips, or ICCs, are smaller and less expensive than the integrated circuits used in third-generation computers. So, the fourth-generation of computers are even smaller and less expensive than the third-generation computers. Computers built with these chips can perform over 10 million calculations in one second. They are 10 times faster than third-generation computers. (That's 1000 times faster than second-generation computers and 10,000 times faster than first-generation computers!)

1st Generation	2nd Generation	3rd Generation	4th Generation
Vacuum tubes	Transistors	Integrated circuits	Integrated circuit chips
1,000 calculations per second	10,000 calculations per second	1,000,000 calculations per second	10,000,000 calculations per second

As you can see, each generation of computers used a new invention to conduct the electricity through the computer. As the new electrical devices got smaller, the computers got smaller. They became more powerful than the earlier, large computers. They also became less expensive, which enabled small companies, schools, and people to buy their own computers.

Lesson 3

COMPUTER COMPONENTS

PERFORMANCE OBJECTIVES: Students will be able to:

- identify the four major computer components:
  - CPU
  - input devices
  - output devices
  - storage (internal and peripheral)
- explain the functions of each of the major components.

VOCABULARY: NEW: central processing unit, input, internal, output, peripheral, storage

REVIEW: CRT, keyboard, printer

MOTIVATION: Present the following issue:

- a Volkswagon, a Mercedes, a Chevrolet, and a Cadillac are all considered automobiles, although they are very different. Why is it possible to call each of them an automobile?
- Discuss basic elements shared by all and the differences which would distinguish a simple "Bug" from a "Caddy."
- Just as with cars, some computers are more "luxurious" than others. Yet all computers, big or small, must have certain essential parts.
- What will we be attempting to learn in this lesson? (Elicit AIM.)

AIM: What are the "parts" that all computers must have?

DEVELOPMENT:

1. Present the following brief exercise:
   - Write this sequence of numbers on chalkboard:

     8, 15, 83, 22, 47, 16, 52, 71, 10, 37

- In your notebooks convert this list into two columns, one for the "odd" numbers and one for the "even" numbers.

- Call for volunteer to place correct answer on board.

2. Many years ago, people used to call computers "mechanical brains"! They were both right and wrong. Let's see how our own brains performed our exercise.

   - What steps did your brain follow to obtain the result?

     - It considered each number separately.

     - It decided if numbers are "even" (divisible by two with no remainder) or "odd."

     - It placed the number in the appropriate column.

3. Let's examine this more closely. Your brain actually performed three separate tasks:

   - It "took in" information. What was this information? (Number to be classified.)

   - It performed a simple calculation. What calculation? (Divided number by two, checked for remainder, classified number based upon calculation.)

   - It produced a result for others to see. What result? (Instructed hand to write number in appropriate column.)

4. It is possible to give formal names to these three operations. Match the operation below to its name:

   (correct responses)

Input	Brain receives information
Process	Brain uses information in calculation
Output	Brain causes hand to write answer

5. Computers also perform the operations of Input - Process - Output. Therefore, it is not totally wrong to think of them as "mechanical brains" although no one uses that term today.

   - Let's find out the technical names for the parts of the computer which perform these operations.

6. When you divide by two, when you decide which item of clothing matches with another, or when you recognize an old friend, your brain is "performing a process."

   - Which of these computer parts would be in charge of performing a process: keyboard, screen, printer, central processing unit? (Central processing unit, or CPU.)

   - Explain that CPU is the "mind" of the computer. It is in charge of arithmetic, logic, decision-making, etc.

   - Computers once filled entire rooms. Today they can fit on a desktop. They are called microcomputers. what does the prefix "micro" usually mean? (small, very small - microscope, microbe, etc.)

   - Inside the microcomputer is the microprocessor. It is smaller than your fingernail. It is the brain. What is its function? (To perform the various processes.)

7. We've identified the "brain" of the computer, the CPU. How did your brain know which numbers to test for "odd" or "even"? (Eyes sent image to brain.)

   - What is the term for information sent in to the brain? (input).

   - How might a CPU get its input? (from the keyboard)

   - Explain that the keyboard is one way of sending information to the CPU. Mention others: card readers, disk drives, even telephone hookups (modems).

   - What do all of these devices have in common? (All provide input to CPU.)

8. Your eyes input information to your brain and your brain decided if the number was "odd" or "even." What was the last operation performed? (Writing numbers in appropriate column.)

   - What is the technical term for putting information out? (output).

   - Where does a computer display its output? (CRT, printer).

9. We have thus far identified three essential parts of the computer. What are they? (input devices, CPU, output devices).

    - There is still one component remaining. Let's discover what it is.

        - Answer the following questions:

            - How old is your mother?

            - What street do you live on?

            - What are three major league teams?

            - What is the capital of the United States?

    - How did you know the answers so easily? (They are in our memory.)

    - What is memory? (A place in the brain in which information is stored.)

    - What does your brain do with the stored information? (uses it for various operations)

10. Computers have memory also. Computer memory is called storage. In this memory the computer stores information and/or instructions that make it work. There are two kinds of storage.

    - One type of storage is actually inside the computer. What might it be called? (internal storage).

    - A second type is actually outside of the computer. What might it be called? (external storage)

    - Explain that another name for external storage is "peripheral" which means "beyond the boundaries" (i.e., outside).

11. Let's discover why there are two types of storage.

    - Why do people consult reference books such as dictionaries, encyclopedias, etc.? (They cannot retain all facts in their brains.)

    - Again, computers are similar to people. Their memories are also limited in size! Engineers discovered a way to store information outside the computer. The computer can thus "take" the information from these storage devices. What external storage devices have you seen? (floppy disks, cassette tapes, etc.)

12. Can you recall examples of an external storage device providing information required to perform a process? (computer games stored on cassette or disk, Atari and other video-game cartridges, etc.)

13. What might happen if you wrote in an encyclopedia or library book? (You would be accused of defacing public property.)

    - Computer storage devices are special. Not only can the computer take in information from them, but it can put information out to them. What are the technical terms for taking in and putting out information? (input and output)

    - Since these external storage devices allow information to be taken in by the CPU and put back out, what might they technically be called? (input/output devices. Emphasize that cassettes and floppy disks can serve both functions.)

14. We have just made computers less mysterious. If you understand input - process - output and storage, you understand how computers work.

APPLICATION:

- Identify each of the following as an example of either input, process, output, or storage:
  - You read a letter written by your vacationing friend.
  - You correctly fill in the blanks on a job application.
  - You memorize the names of the first thirteen colonies.
  - You calculate how much money you earned in the first week of your new job.
  - You decide what to buy your mother for her birthday.
  - You sign your name to the birthday card.
  - You multiply your age by the first three digits of your telephone number.
  - You add that number to the combined ages of your family.

- You write the answer on a blank sheet of paper.
- You memorize the phone number of that new special someone.

SUMMARY:

- Which devices supply information to the "brain" of the computer?
- What is the term for information coming "in"?
- What is the abbreviation for the "brain" of the computer?
- Why might the CPU be called a micro-processor?
- How does the computer let a human see results of its calculations?
- What is the technical term for the putting out of such information?

EXERCISES (ASSIGNMENTS):

- You are taking a computer-scored test. You are using a special answer sheet on which you indicate your choices by darkening in a space with pencil. Once the answer sheets are collected they are loaded into a special machine which can "read" the pencil marks into a computer. The computer counts the number of right and wrong anwers for each student and prints out a list of student names and their scores. Describe what just happened in terms of input, for process, and for output.
- Besides on paper, where else might the company save the scores for the entire class? Why couldn't the scores simply be kept inside the computer? Explain in a paragraph.

Lesson 4A

COMPUTER CAREERS I — CREATING CAREER CLUSTERS

PERFORMANCE OBJECTIVES: Students will be able to:

- create a list of careers directly related to the various uses of the computer in the world of work.
- create a list of careers which, though not computer-specific, have been indirectly affected by the development of computers.
- devise career categories (clusters) and allocate the previously named careers into the appropriate clusters.

VOCABULARY: NEW: category, cluster

REVIEW: computer programmer, salesperson, service technician, assembly-line worker, designer, trainer, manual

MOTIVATION:

- Conduct an open but orderly brainstorming session in which the class is encouraged to have recorders fill the entire chalkboard with their suggestions for computer-related careers.
- Let's have a brainstorming session. You will have several minutes to think of and to write down every job or career in any field that is in any way related to computers. Do not prejudge your answers. If you think of a career, write it down.
- Allow adequate time for the above. Obtain volunteer recorders who will write at chalkboard, and a secretary who will transcribe board results onto paper.
- We have filled the board with our suggestions. Many of these are similar. Is there any way we can organize our list by grouping similar ideas? (Elicit AIM.)

AIM: How can we learn more about computer careers by creating career clusters?

DEVELOPMENT:

1. Scan the board and tell us how many of these careers are similar because the worker uses a computer or computerized device in the daily performance of the job.

   . As suggestions are offered, the recorders should circle the items named. The secretary will begin list of the named items.

   . What do all of these workers share in common? (They operate a computer or computerized device, i.e., word processor.)

   . We just grouped similar items together. What is the term for a group of related items? (Category, cluster.)

   . Each category should have a unique name. What shall we name this? (People Who Work With Computers or Computerized Equipment.)

   . Once the secretary has copied and labelled this cluster, have recorders erase circled items from board.

2. The computer may seem like a genius, but it is just a machine. It requires people to make it work. If our first cluster was "People Who Work With Computers" then what shall we name our next cluster? (People Who Make the Computer Work.)

   . Repeat list-making process as above. Acceptable answers would include: Computer Programmers, Computer Service Technicians, Computer Installers, etc.

   . Secretary copies new cluster. Recorders remove circled items.

3. How many different brands of computers can you name? (Attempt to develop an inclusive list, from the most common such as Timex to the more exotic, such as Televideo, etc.)

   . Thus we can see there are dozens of different companies involved in making computers or computer equipment. Let's pretend that you own a company that wants to introduce a new computer.

   . What types of experts and workers do you need? (Designers, engineers, assembly line workers, technicians, etc.)

- List above suggestions on board (or assign recorders to do so). We have just created a new cluster. What shall we name it? (People Who Create and Build Computers and Computer Equipment.)

- The secretary should copy.

4. Your new company wants to sell its computers. How are most computers or other products sold? (In stores, specialized stores.)

   - How would you let the world know that you had a product for sale? (Advertise.)

   - We have just created a new cluster, called People Who Market Computers and Computer Equipment. What careers might it include? (Computer salesperson, advertising writer, advertising artist.)

5. Computers are fun, but to use them well requires some training. How might a person be trained to use a new computer? (Taking classes, reading books, reading the manual.)

   - Here is another cluster. What might we call this cluster of careers? (People Who Explain the Computer to Others.)

   - What careers might be included in this cluster? (Computer teacher, trainer, computer textbook writer, computer manual writer, etc.)

   - Secretary should copy the new cluster.

6. All companies, whether they make computers or grapefruit knives, need certain workers who operate behind the scenes to help the company do its job. Who are some of these behind the scenes workers? (Secretaries, accountants, clerical workers, transportation workers, shipping department workers, etc.)

   - Let's follow some logic. If there are many new computer companies growing each year, how will this help the employment chances for these "behind the scenes" workers? (There will be a need for more of these workers.)

   - How is it possible to have a job in the computer field even if you know very little about computers yourself? What might we call this final cluster? (People Who Help Computer Companies to Operate.)

   - Have secretary record this final cluster.

7. The secretary will come to the board and list the following Computer Career Clusters. The class will copy into notebooks for future reference.

Computer Career Clusters

- People Who Work With Computers or Computerized Equipment
- People Who Make the Computer Work
- People Who Create and Build Computers and Computer Equipment
- People Who Market Computers and Computer Equipment
- People Who Explain the Computer to Others
- People Who Help Computer Companies to Operate

. This list will be necessary for the application below.

APPLICATION:

- Distribute "Computer Career Cluster Worksheet." (See attached. Teachers must duplicate in advance.)
- Review instructions on the worksheet. Answer questions and clarify. Allow time for class to complete first segment of worksheet. Provide for review to ensure that correct answers are entered.
- Explain that the worksheet is to be completed as part of the exercises for this lesson.

FINAL SUMMARY:

- Give five examples of careers that the computer has helped to create.
- What is the purpose of clustering computer careers?

EXERCISES (May be given as homework):

- Complete the Career Cluster Worksheet as explained above.
- Select the one career that you would most enjoy. In a well-written paragraph of six to eight sentences, describe why you would enjoy this career, how its suits your abilities, etc.

BEST COPY AVAILABLE

WORKSHEET

Instructions: Below you will find seven brief descriptions. Each description is associated with one of the Computer Career Clusters. Read each description and decide to which of the clusters it would best belong. Explain your decision in complete sentences.

I. John J. couldn't sleep. Today was the big day -- the day when the QUAD-IV would be unveiled to the top executives of the company. John's department had spent almost all of the past year designing and building this super-computer. His whole department believed that the QUAD-IV would surpass any other computer now available. If the public agreed, the company would make a fortune. But before any of this could happen, John would have to convince the company that the new computer was worth producing and that it wasn't just an experiment.

II. Maria tried hard to avoid shouting. Walking out the door now was the purchasing officer of a major New York City bank. He had just signed a contract with Maria's company. The bank would be spending almost a quarter-million dollars to buy computer equipment from her company, and Maria would be getting a huge sales commission for getting Mr. Smith to sign on the dotted line. She could hardly control her excitement, but she had to remain calm. A customer was walking toward her desk. Maybe this would be another sale.

III. The clock on the wall announced 5:35 p.m. Long past quitting time for most people, Anthony thought. But Anthony wasn't "most people." Neither was his entire department. All of them were staying late, sitting at their desks with pencils in hand or typing awav at computer terminals. The payroll program was not working properly, and they all knew that no one could leave until the problem had been corrected. Anthony had narrowed down the problem to one particular part of the program. Now he tapped his pencil against his teeth as his mind wrestled with a way to clear the jam-up and get the program working again.

IV. Sarah was amazed. She had been an accountant for five years, but she couldn't help being impressed by the amount of money that her company had earned in just the past six months. The company was rather small, and all it sold was computer paper and other computer supplies. Still, she thought, there must be a tremendous market for their product. They've been doing better each month. In fact, the tiny company had grown so fast that they had to hire twenty new employees. Also, they had opened a new office with two full-time accountants. Sarah was just as important to the company as the folks on the production line.

V. "Where is it?" Nancy asked the receptionist. The woman pointed to a door marked "Data Center." Opening the door, she spied a pair of legs protruding from behind one of the high-speed printers. "This is no place for a do-it-yourself job," she said as a way of greeting. From behind the waist-high machine appeared the shape of the data-processing manager. Brushing dust from his suit, he smiled and said, "Glad you got here so fast. This printer has been acting strange all week, and today it just stopped. Wouldn't you know that we need it for a huge job tomorrow?" "That's why I'm here," smiled Nancy. "So move over and let an expert take a look at it!"

VI. Nick breathed a sigh of relief. The procedure had taken longer than usual, but the results coming off the thermal printer showed that everything was normal. The printer was attached to a computer console, and that was attached to a huge apparatus which could produce computerized three-dimensional X-rays of a patients's entire body. The public knew the machine as a CAT-scanner, but Nick and the other technicians knew it as a miracle. The equipment saved dozens of lives, for it could show doctors exactly where in a patient's body a tumor might be hiding. This time, though, there would be rejoicing, for the machine determined that his patient could go home, free from worry about tumors. It was times like this that made Nick glad he had decided to train to enter the field of computerized medical diagnosis. Without computers, Nick thought, people would die needlessly.

VII. "Those students are really learning fast," thought Sandy. She was looking over the latest set of exam papers from her computer literacy class, and she was very pleased. Most of the students had not known a thing about computers when they registered for the course. Now some of them were writing programs that would please an expert. More important, she thought, they are enjoying the course, and several are talking about becoming computer-science majors. She felt good, for she knew that her teaching had made the computer less mysterious to another group of people.

Lesson 4B

COMPUTER CAREERS II — QUALIFICATIONS AND TRAINING

PERFORMANCE OBJECTIVES: Students will be able to:

- indicate the specific abilities required for successful performance of jobs in the Computer Career Clusters.
- recommend specialized training required for success in these careers.

VOCABULARY: NEW: qualifications, abilities

REVIEW: Computer Career Clusters

MOTIVATION:

- What different types of workers can be found in an average school? (Teachers, assistant principals, principal, guidance counsellors, secretaries and clerical workers, custodian and staff.)
- Let's create a career cluster for each of these workers.
    - People who help others to learn subject.
    - People who organize and run the school.
    - People who help students with personal problems.
    - People who handle daily clerical chores.
    - People who keep physical equipment functioning.
- Review the concept of a cluster as a category of similar ideas.
    - Ascertain that students understand how various school personnel can be placed into these clusters.
- Each of these clusters requires special qualificatons or abilities from the workers doing the jobs.
    - What abilities would be required for the first cluster? (Ability to present subject matter clearly, patience, kindness, ability to enjoy people, etc.)

- What abilities would be required for the second cluster? (Ability to organize, take charge, supervise, etc.)

- You may want to proceed as above for all five clusters. Otherwise, continue as follows.

- Each of these workers had to obtain special training. Let's see what training some of them had.

  - What training might be required for People Who Help Students With Personal Problems? (Training in psychology, adolescent problems, etc.)

- What training would be needed for People Who Keep Physical Equipment Functioning? (Mechanical training, training in diagnosis and repair, etc.)

- We have demonstrated that various jobs require various abilities and training. What do we wish to learn about the various computer careers? (Elicit AIM.)

AIM: What abilities and training are needed for the Computer Career Clusters?

DEVELOPMENT:

1. Distribute copies of Worksheet II. The teacher should reproduce sufficient quantities beforehand. Allow adequate time for students to familiarize themselves with this worksheet.

2. The worksheet contains each of the Computer Career Clusters developed in the previous lesson. Below each cluster is space in which students will write specific abilities and training. At the bottom of the worksheet can be found two lists: one for abilities and the other for training. Have students scan these lists and allocate appropriate abilities and training to the various Computer Career Clusters. Encourage students to be imaginative, as there will be opportunity for discussion of their findings later in the lesson.

3. You may wish to have class do the first cluster together, as means of demonstrating the method to be followed. Allow adequate time for class to complete remainder of worksheet.

4. While class is working, reproduce the upper segment of sheet on chalkboard. Call for and select volunteers to serve as recorders and secretary. Recorders will copy student responses onto chalkboard. Secretary will keep official record of responses on a fresh worksheet.

5. Call for answers to complete worksheet copied onto chalkboard. Remind students that all responses are worth considering and that there are no right and wrong answers here. If an answer can be justified, it is correct.

6. Why is it important to identify the abilities required for a particular Computer Career Cluster? (To determine whether one is suited for that career.)

   - What should a person do if he/she realizes that he/she does have the requisite abilities? (Find out more about our careers, speak with guidance counsellor, plan to enroll in appropriate high school level courses, find out how to obtain specialized training, etc.)

7. What is the purpose of the activity in this lesson? (To learn more about careers in computer area, to help us to evaluate ourselves, to "raise our sights," etc.)

   - This lesson also serves to introduce the special long-term Career Project. Introduce project by distributing assignment sheet now, if possible.

FINAL SUMMARY:

- Why is it important to understand the requirements of a career?
- Why is it important to evaluate our specific abilities?
- Why is it important to do further research before deciding upon one of these careers?

EXERCISES (ASSIGNMENTS):

- Introduce Long-Term Career Project. See Project Assignment. (This must be reproduced in sufficient quantity.)

- Pretend that you are author of an advice column, such as "Dear Abby." Write an intelligent and serious answer to either of the following letters:

  - Dear Advice Person:

    I am a high school student in search of a career. I enjoy being around people and helping them. I am fairly good with my hands and enjoy learning how things work. I get relatively good marks in math and in science. However, I do not want to work in the same office every day. I enjoy getting out and meeting people. Can you suggest a career in the computer field?

    -signed, Moe Demn

  - Dear Advice Person:

    I am miserable! I have always wanted to be a computer programmer, but my grades in math are not good. I enjoy solvng problems, and I get really involved with a good crossword puzzle. I think computers are exciting and I've always dreamed of working with them, but it doesn't look promising. What should I do?

    -signed, Failure at Fourteen

WORKSHEET

Computer Career Clusters -- Abilities and Training

Instructions: Below each of the Computer Career Clusters is a separate space in which you may indicate the specific abilites and training needed for that particular cluster. At the bottom of the sheet appears listing of the various abilities and training. Remember: if you can provide a good reason then your answer is correct. There are no "wrong answers."

Note: Any item from the Abilites and Training list may be used as many times as necessary. None are ever eliminated.

- People Who Work With Computers and Computerized Equipment:

  Abilites Needed:

  Training Required:

- People Who Make the Computer Work:

  Abilites Needed:

  Training Required:

- People Who Create and Build Computers and Computer Equipment:

  Abilites Needed:

  Training Required:

- People Who Market Computers and Computer Equipment:

  Abilites Needed:

  Training Required:

- People Who Explain the Computer to Others:

  Abilites Needed:

  Training Required:

- People Who Help Computer Companies to Operate:

  Abilites Needed:

  Training Required:

....

Abilities and Training Section

Select from these to complete the worksheet. You may also add any not included.

Abilities:

- to pay attention to detail
- to solve problems and to think logically
- to explain ideas to others in clear fashion
- to work with sophisticated equipment
- to handle routine work without making errors
- to work alone
- to revise and recheck work until it is perfect
- to follow directions
- to comprehend electronic circuitry
- to convince people
- to perform clerical tasks
- to deal with the public
- to find short-cut solutions to complex problems
- to work under the pressure of a deadline
- to accurately handle technical equipment
- to perform secretarial duties
- to work at a job until it is completed

Training:

- in electronics
- in computer programming
- in psychology of people
- in mathematics and science
- in clerical and office skills
- in computer systems
- in written expression

Lesson 4C

COMPUTER CAREERS III — A LOOK AT THE FUTURE

PERFORMANCE OBJECTIVES: Students will be able to:

- state the various ways in which the computer may revolutionize methods in commerce and industry.
- cite certain jobs which may be altered or eliminated by computers.
- explain how retraining in computer skills may help workers to avoid becoming obsolete in the job market.
- state what steps a student should take now to guarantee future employment in a computer-intensive work force.

VOCABULARY: NEW: automation, electronic mail, robotics, telecommunication

REVIEW: microprocessor

MOTIVATION:

- Distribute sheet and read aloud the words to "Allentown" by Billy Joel (Duplicate sheet in sufficient quantities for distribution to class.)
- What might have inspired this song? (Steel factories closing, workers being laid off.)
- How many of you know families in which the breadwinner has been laid off? What were some of the reasons for these layoffs? (Poor economy, company relocating, etc.)
- Some people say that the invention of computers is partly responsible for loss of jobs. How is that possible? (Discuss automation and its effect on work force.)
- Others say that the computer is actually creating new jobs. How might that be possible? (New companies in computer field need more workers, etc.)
- (Elicit AIM.)

AIM: What effect might computers have on employment now and in the future?

DEVELOPMENT:

1. Distribute attached worksheet. (Reproduce adequate quantities for distribution.)

   - Allow adequate time for students to familiarize themselves with the document.
   - Worksheet contains a list of various jobs, some of which will probably experience strong growth patterns while others will probably suffer declines.
   - Arrange the class into groups. Each group will have the responsibility to study the worksheet and attempt to predict which jobs will grow and which will decline by the start of the next century.
   - Allow a reasonable amount of time for completion of task.

2. While the class is working in committee, create a chart on the chalkboard with following column headings:

   By the Year 2,000

Careers Which Will Grow Stronger	Careers Which Will Decline

   - Select volunteers to record responses on the board and an official secretary who will maintain a paper copy of the results.

3. Have students present the findings. Begin with the "growing" column. Emphasize that, as might be expected, careers involving computers will experience strong patterns of growth.

   - Allow committees to present the findings for the column showing growth. Remind the class that this is, of course, only speculation based upon past trends.

4. Complete the exercise by obtaining the findings for declining careers. Elicit reasons as follows:

Careers Which Will (Probably) Decline

Career	Possible Reason
Bank Teller	Electronic Banking (automatic tellers)
Assembly Line Worker	Robotized Assembly Lines
Bookkeeper	Computerized Accounting Systems
Supermarket Checker	Computerized Scanners (bar-code)
Department Store Clerk	Shop-at-Home Systems (Qube)
Librarian	On-line Information Retrieval
Telephone Operator	Computerized Switching Equipment
File Clerk	Computerized File Maintenance

5. Guide a discussion based on questions such as the following: If you were a worker in one of these declining areas, how might you feel about this trend? (angered, scared)

   - Explain the derivation of the word "sabotage.") (Workers in France faced with loss of jobs caused by the 19th Century Industrial Revolution attempted to retaliate against this early automation by throwing wooden shoes, called "sabots" into machinery to destroy it. Hence:

     Sabotage - (n), "malicious injury to machinery... Any underhanded interference with production..."

6. Why is sabotage not an adequate solution to this problem? (possibility of imprisonment, cannot stop progress, etc.)

   - What would be a more acceptable solution? (Retrain for jobs in the growing areas.)
   - What sort of training should be sought? (Computers, computerized equipment, etc.)
     - Note: allow for debate, as this may not be a completely viable solution for all concerned.
   - Explain that while no one can really predict the future, it is important to think seriously about trends which will influence the future. What messages are there for a young person still in school today? (Computers are here to stay. Computers will play an increasingly important role. Those people with computer-related skills will probably have a better opportunity for employment. Therefore, it would be wise to learn as much as possible about computers and related areas.)

FINAL SUMMARY:

- How is it possible for the computer (microprocessor) to both help and harm the employment situation?

- How might a young person take advantage of this trend of computerized automation in the workplace?

EXERCISES (ASSIGNMENTS):

1. Interview a parent, relative, or friend who works full-time. Ask the following questions, record your answers, and be prepared to report to the class at a future date:

   - What work does this person do?
   - What specific signs have there been that computers and/or microprocessor-based equipment are having an effect on employment, either positive or negative?
   - If jobs have been lost to automation, what has been the reaction of the workers still employed?
   - What would this person do if he/she learned that his/her job was about to be allocated to computerized equipment?
   - Is this person willing to retrain for jobs in the new technology? Why or why not?

2. In a well-constructed paragraph, describe how you could make yourself "automation-proof"; i.e., what you can do to be sure that you will not lose a job because of the new technology. Give specific examples of skills you might need to acquire, subjects to be taken, etc.

Song Sheet

(To be distributed as a part of motivation segment)

ALLENTOWN

Well we're living here
    in Allentown
And they're closing all
    the factories down
Out in Bethlehem they're
    killing time
Filling out forms
Standing in line
Well our fathers fought
    the Second World War
Spent their weekends on
    the Jersey Shore
Met our mothers in the USO
Asked them to dance
Danced with them slow
And we're living here in
    Allentown

....

Well we're waiting here
    in Allentown
For the Pennsylvania we
    never found
For the promises our teachers
    gave
If we worked hard
If we behaved

....

And we're waiting here in
    Allentown
But they've taken all the
    coal from the ground
And the union people
    crawled away
Every child had a pretty
    good shot
To get as least as far
    as their old man got
But something happened on
    the way to that place
They threw an American
    flag in our face

"Allentown"
words and music by Billy Joel
from The Nylon Curtain album
(38200)
1982 Columbia Records
CBS Inc.
51 West 52nd Street
New York, New York

WORKSHEET

Jobs which May Grow vs. Jobs which May Decline

Below is a list of jobs. Although no one can predict the future, it is possible to make intelligent guesses based on recent trends in our world.

It is your task to scan the list of jobs. Mark with a "G" those which might experience a pattern of growth by the year 2,000, and mark with a "D" those which might decline in number by the year 2,000.

Do not hesitate to make a judgment. There are no right or wrong answers, for we will have to wait until the year 2,000 to find out.

Careers

Bank Teller
Computer Repair Technician
Automobile Mechanic
Newsstand Operator
Data-Entry (Computer Terminal) Clerk
Telephone Operator
Weather Forecaster
Assembly Line Worker
Computer Salesperson
Department Store Clerk
Bookkeeper
File Clerk (paper)
Electronics Plant Assembly-Line Worker
Computer Programmer
Supermarket Checker
Electronics Technician
Mechanical Typewriter Repairperson
Computer Operator
Secretary
Receptionist
Air Traffic Controller

Lesson 5A

KEYBOARD SKILLS — TRS-80

PERFORMANCE OBJECTIVES: Students will be able to:

- explain the use of selected regular and special-function keys.
- use these keys to achieve the results for which they are intended.
- demonstrate an understanding of the concept that the keyboard is the primary vehicle through which the user communicates with the CPU.

VOCABULARY: NEW: Function, Cursor, Space-bar, Upper-Case, Lower-Case, Upper-Lower Case

REVIEW: Screen, Keyboard

MOTIVATION:

Elicit responses to the following:

- How does a driver control a car?
- How does a pilot control an airplane?
- How does a captain control a ship?
- How do you operate a touch-tone telephone?
- How does an office worker operate a mimeograph machine, adding machine, etc?

Develop concept that various kinds of equipment, from simple telephone to a 747, must be operated and/or controlled by a person using devices such as buttons, switches, wheels, levers, etc.

- The advanced technology that we have been studying is the computer. What might we be trying to learn in this lesson? (Elicit AIM).

AIM: What are the special controls that we use to operate the computer?

DEVELOPMENT:

1. How is our computer similar to a typewriter?
   (Both have keyboards which enable us to operate the device.)

2. Computers and typewriters both have keyboards, but the computer has special keys to do special jobs. Let's look at some of these. Both have a special-shaped key called a Space-bar. What happens when you press the space-bar of a typewriter?
   (Carriage moves ahead one space.)

3. Find the space-bar on the computer keyboard. Press it once while you watch the screen. What happened?
   (Rectangle of light moved.)

4. What is the rectangle of light called?
   (Elicit or state that it is called a "Cursor" which is Latin for "Runner.")

5. The cursor moves when you press the space-bar. Make it move ten times. (Students will press space-bar ten times.)

6. Recall that we said the computer can do more than a typewriter. Let's find the key marked → (right arrow). Press it once. What happened?
   (Cursor moved ahead 8 spaces. Repeat to prove.)

7. Find the key marked ← (left arrow). Press it once. What happened?
   (Cursor "backed up" one space.) Repeat to prove.

8. What is the difference between the way the right and left arrow keys function? (Right arrow advances cursor 8 spaces each time. Left arrow moves cursor back only one space at a time.)

9. What have we typed on the screen so far?
   (Nothing or spaces.) (Spaces are a form of output.)

10. How could we make letters or numbers appear on the screen?
    (Press appropriate keys.)

11. Please press the key labelled "Clear." Now type your initials. Watch the screen. Where is the cursor now? (Cursor appears immediately after last letter on screen.)

12. Type a space and watch the screen. What happened? (Space appeared, followed by cursor.)

13. What might the position of the cursor tell the user? Where the next letter/number will appear when typed.)

14. Make your initials appear five spaces away from last letter on screen. (Student will press space-bar four times then type initials.)

15. Please press the key labelled "Clear." What happened? (Screen displayed disappeared. Cursor now at start of top line.)

16. We will now type our complete names, leaving one space between words. How might we erase what we just wrote? (Use left arrow key to "back-up" and erase word.)

17. There is an easier way to erase an entire line. Try this: Press "Clear." Re-type your full name. Now hold down the "Shift" key and press the left arrow key. What happened?
(Shift + left arrow erases entire line and returns cursor.)

18. We've just discovered that the computer has special controls that let you do much more than you could with a typewriter. Let's start a list in our notebooks of these "special-function" keys:
(Elicit and place following chart on board):

KEY(S)	FUNCTION (JOB) PERFORMED
Space-bar	Moves cursor ahead once each time.
Clear	Clears screen, returns cursor.
——>(right arrow)	Moves cursor 8 spaces forward.
<—— (left arrow)	Backspaces and erases once each time.
Shift + (left arrow)	Erases entire line, returns cursor.

19. We will be adding to this chart, but let's get some practice using what we have learned. Clear the display.

- Type "I am learning to control the computer."
- Erase the last letter.
- Erase the last word.
- Erase the entire line.

20. Re-type the same sentence, but put two spaces between each word. Now continue to tap the space-bar and watch the cursor. What happened when the cursor reached the right-hand edge of the screen?
(Cursor jumped down to start of next line.)

21. Examine your keyboard. Is there a key that might make the cursor move down to the next line?
(Down arrow.)

22. Press the down arrow key five times. Where is the cursor now? Continue to press this key until the cursor will not go down any further. What is happening on your screen?
(Display is moving off top of screen. Cursor remains stationary.)

    - Explain: The screen has a fixed number of 16 lines. To add additional lines, the top line is moved upward and is "lost." (Up-arrow will NOT move cursor upwards.) Pressing "Clear" will clear the video display and move the cursor up.

23. We are now ready to have some fun with our computer.

    - Clear your screen.
    - Press Shift + right arrow.
    - What happened to the cursor?
      (It became bigger or wider.)
    - Type your first name. What happened?
      (Letters appear to be "double-width.")
    - Leave one space after your name and add "is learning to use the keyboard."
    - What happened on the screen?
    - (Sentence was continued on next line.)
    - Why did such a short sentence take up so much room?
      (Each letter is now double-width. Thus half as many letters will fit on a line.)

24. Please press "Clear." Type the same sentence. Why did it fit on one line now?
(Pressing "Clear" changes display from double to normal width in addition to moving the cursor up.)

25. Let's learn one more trick with our computers. Press "Clear" again. Type this sentence exactly as shown on board, "BOOKS DON'T LOOK LIKE THIS." Why don't they?
(Above sentence is all capital letters or upper-case. Books are printed with a mixture of capital and small letters or upper-lower case.)

26. Let's make the screen display look more like the printed page.

- Hold down Shift key and press zero. Now type the exact same sentence. Why does it still look strange?
(They are all small or lower-case letters.)

- How could we add capital letters to each word?
(As on typewriter, the Shift key must be held down to obtain capitals.)

27. Type this sentence, but make each word start with an upper-case letter:
"The Computer Uses Upper And Lower Case If You Know The Secret."

28. Hold down the Shift key and press zero again. Now type, "HEY, WHAT HAPPENED?" What did happen?
(Display is now once again in upper-case mode.)

What is the function of Shift + 0 (zero)?

(It switches in and out of upper case or lower case mode. It provides an either/or function, like a toggle switch.)

29. We've learned a lot more about controlling the computer. Let's turn to our chart again:

KEY(S)	FUNCTION (JOB) PERFORMED
↑ (down arrow)	Moves cursor to beginning of next line.
Shift + —>(right arrow)	Creates double-width letters.
Clear	Returns to normal-width (clears display).
Shift + 0 (zero)	Switches upper-case to lower-case and vice-versa.

FINAL SUMMARY:

- How are computer keyboards similar to typewriter keyboards?
- How are computer keyboards different from typewriter keyboards?
- What do we call that group of keys which allows us special control over the display?
- What is the official name of the blinking block of light? What is its purpose?
- Describe what happens when a user presses each of the following keys or combination of keys:
    - Clear
    - Space bar
    - right arrow →
    - left arrow ←
    - down arrow ↓
    - Shift + ← (left arrow)
    - Shift + → (right arrow)

APPLICATION:

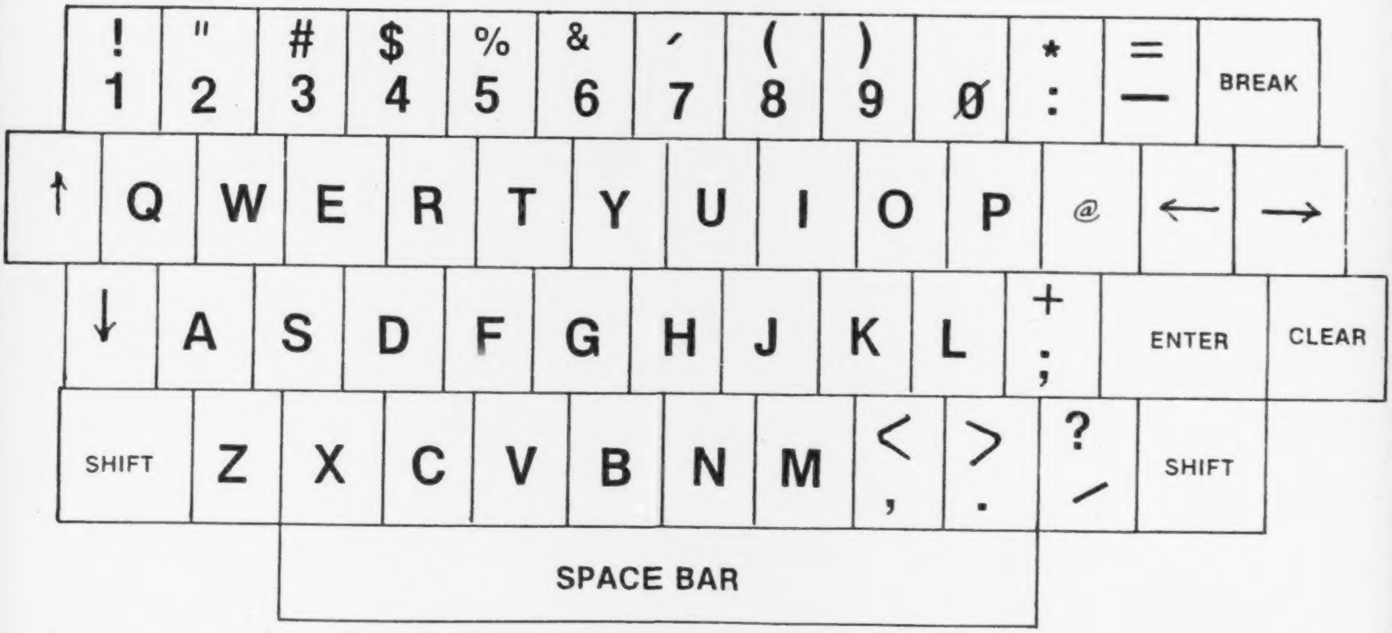
- Clear display. Make the following sentences appear exactly as shown:
    - THESE LETTERS ARE TOO BIG, SAID GOLDILOCKS!
    - these letters are too small, said goldilocks!
    - These letters are just right, said Goldilocks!
- Return to upper-case. Do the following:
    - Type "FAT CATS ARE FUN" in double-width.
    - Bring cursor down two lines.
    - Type "SKINNY CATS ARE FUN ALSO" in normal-width.
    - Type "this is fun" in normal-width, all lower-case.
    - Bring cursor down one line, type "This is more fun" in double-width.
- Cursor-Dance.
    - Clear display.
    - Place a line of 30 dots on your screen.
    - Erase last five.
    - Erase five more.
    - Erase entire line.
    - Bring cursor down two lines.
    - Type your first name.
    - Bring cursor down one line.

  - Make cursor move 16 space ahead.
  - Type last name.
  - Make cursor move 24 spaces ahead.
  - Type any three words.
  - Erase entire last line.
  - Retype it.
  - Type "I am done!"

- Surprise Message.

  - Clear display.
  - Type "SAMTINLEED!" exactly as shown.
  - Hold down Shift key and press right arrow. What happened?
  - Display changed to "SMILE!" because double-width mode absorbs two spaces for each letter, obliterating "A, T, N, E, D" in SAMTINLEED!
  - Have students design their own surprise messages.

TRS 80


! 1
" 2
# 3
$ 4
% 5
& 6
' 7
(8
) 9
0
* :
= —
BREAK
↑
Q
W
E
R
T
Y
U
I
O
P
@
←
→
↓
A
S
D
F
G
H
J
K
L
+ ;
ENTER
CLEAR
SHIFT
Z
X
C
V
B
N
M
< ,
> .
? /
SHIFT
SPACE BAR

Lesson 5B

KEYBOARD SKILLS — PET

PERFORMANCE OBJECTIVES: Students will be able to:

- Explain the use of selected regular and special-function keys.
- Use these keys to achieve the results for which they are intended.
- Demonstrate an understanding of the concept that the keyboard is the primary vehicle through which the user communicates with the CPU.

VOCABULARY: NEW: Function, cursor, cursor control, space-bar, upper-case, lower-case,

REVIEW: Screen, keyboard

MOTIVATION:

- Elicit responses to the following:
  - How does a driver control a car?
  - How does a pilot control an airplane?
  - How does a captain control a ship?
  - How do you operate a touch-tone telephone?
  - How does an office worker operate a mimeograph machine, adding machine, etc?
- Develop the concept that various kinds of equipment, from a simple telephone to a 747, must be operated and/or controlled by a person using devices such as buttons, switches, wheels, levers, etc.
- The advanced technology that we have been studying is the computer. What might we be trying to learn in this lesson? (Elicit AIM.)

AIM: What are the special controls that we use to operate the computer?

DEVELOPMENT:

1. How is our computer similar to a typewriter?
   (Both have keyboards which enable us to operate the device.)

2. Computers and typewriters both have keyboards, but the computer has special keys to do special jobs. Let's look at some of these. Both have a special-shaped key called a Space-bar. What happens when you press the space-bar of a typewriter?
   (Carriage moves ahead one space.)

3. Find the space-bar on the computer keyboard. Press it once while you watch the screen. What happened?
   (The flashing rectangle of light moved.)

4. What is the rectangle of light called?
   (Elicit or state that it is called a "cursor" which is Latin for "runner.")

5. Press the space-bar and hold it while you count to three. What happened?
   (The cursor moved more than one space.)
   (On older PETs the cursor only moves once.)

6. Make the cursor move ten times. Be careful not to hold the space-bar down too long.
   (Students will either hold down the space-bar or press it ten times.)

7. Find the key marked "RETURN" and press it. What happened?
   (The cursor returned to left side of screen, one line lower than before.)

8. Remember that we said the computer can do more than a typewriter. Let's find the key marked "HOME." Press the HOME key. What happened?
   (Cursor returned to upper left corner of screen.)
   (The sign-on message should still be there.)

9. Press the space-bar once. What happened?
   (A space appeared and the cursor moved.)

10. Now find the key marked "CRSR" ——> (right arrow.) This is the Cursor right key. Press it. What happened?
    (The cursor moved, but the blank did not appear.)

11. Now find the key marked ← (left arrow.) How can you press it? (It is the same key as CRSR right.) Find the key marked "SHIFT"; there are two of them. Press the shift key and the CRSR left. What happened?
(The cursor backed up one space.)

12. What have we typed so far?
(Nothing, or spaces.)

13. Well, how could we make letters or numbers appear on the screen?
(Press appropriate keys.)

14. Find the CRSR down key. Press it once. What happened?
(The cursor moved down one line.)

15. Move the cursor down to a blank line. Now type your initials. Watch the screen. Where is the cursor now?
(The cursor appears immediately after the last letter on the screen.)

16. Type a space and watch the screen. What happened?
(The cursor moved, leaving a space.)

17. Make your initials appear five spaces away from last letter on screen. (Student will press space-bar cursor four more then type initials.)

18. Press the RETURN key. What happened?
(Computer prints: ?SYNTAX ERROR
READY.)

19. When you press the Return key, the computer tries to carry out the instruction that you have typed. Your initials are not an instruction. The message "SYNTAX ERROR" means "That is not an instruction." Now press the shift key and the key marked "CLR." What happened?
(The screen display disappeared. The cursor is now at the start of the top line.)

20. We will now type our complete names, leaving one space between words. How might we erase what we just wrote?
(Use CRSR left arrow key to "back-up" and erase word.)

21. There is an easier way to erase letters. Press return. Now type your name again. Find the key marked "DEL" and press it once. What happened?
(One letter disappeared and the cursor backed up.)

22. We have just discovered that the computer has special controls that let you do much more than you could with a typewriter. Let's start a list in our notebooks of these special keys:
(Elicit and draw following chart on board):

KEY(S)	FUNCTION (JOB) PERFORMED
Space bar	Moves cursor ahead; makes blank
CRSR right	Moves cursor ahead
Shift+CRSR left	Backspaces (Moves cursor back)
DEL	Backspaces and erases
CRSR down	Moves cursor down.
RETURN	Moves cursor down and to start of next line. Computer tries to follow instruction.
HOME	Returns cursor to top left corner of screen.
Shift+CLR	Clears screen, homes cursor.

23. We will be adding to this chart, but let's get some practice using what we have learned. Clear the display.

    - Type "I am learning to control the computer."

    - Erase the last letter.

    - Erase the last word.

    - Erase the entire line.

24. Retype the same sentence, but put two spaces between each word. Then continue to tap the space-bar and hold it down. What happened when the cursor reached the right-hand edge of the screen?
    (Cursor jumped down to start of next line.)

25. What other key will make the cursor move down?
    (CRSR down, and RETURN.)

26. Make the cursor move down almost to the bottom of the screen. Now watch carefully and make it move down to the bottom. Press the key again. What is happening on your screen?
    (The cursor is stationary, but the display is moving off the screen.)

    - Explain: The screen has a fixed number, 25, lines. To add additional lines, the top line is moved upward and is lost. This is called "scrolling."

27. How might you move the cursor upwards.
    (Use shift+CRSR up, or CLR.)

28. Now try this: Type "A", CRSR down, "B", CRSR down, "C", CRSR down, and so on. What do you see?
    (The letters form a diagonal line.)

29. Press HOME. Now find the key marked "RVS" and press it. Use the CRSR right key to find a blank space on the line. Type "A" and CRSR down. What happened? (The "A" is in reverse with a light background.)

30. Make another diagonal of reverse letters. How might you make the typing normal again?
(Shift+OFF)
(RETURN also turns off reverse.)

31. We've learned a lot about controlling the computer. Let's turn to our chart again.

KEY(S)	FUNCTION (JOB) PERFORMED
Shift+CRSR up	Moves cursor up.
RVS	Prints in reverse.
Shift +OFF	Turns off reverse: prints normally

SUMMARY:

- How are computer keyboards similar to typewriter keyboards?
- How are computer keyboards different from typewriter keyboards?
- What do we call that group of keys which allows us special control over the display?
- What is the name of the blinking block of light? What is its purpose?
- Describe what happens when a user presses each of the following keys or combination of keys:
  - Space bar
  - CRSR right
  - CRSR left
  - RETURN
  - DEL
  - CRSR down
  - Shift + CRSR up
  - CLR
  - HOME
  - RVS
  - Shift + OFF

APPLICATIONS:

- Clear display. Make the following sentence appear:

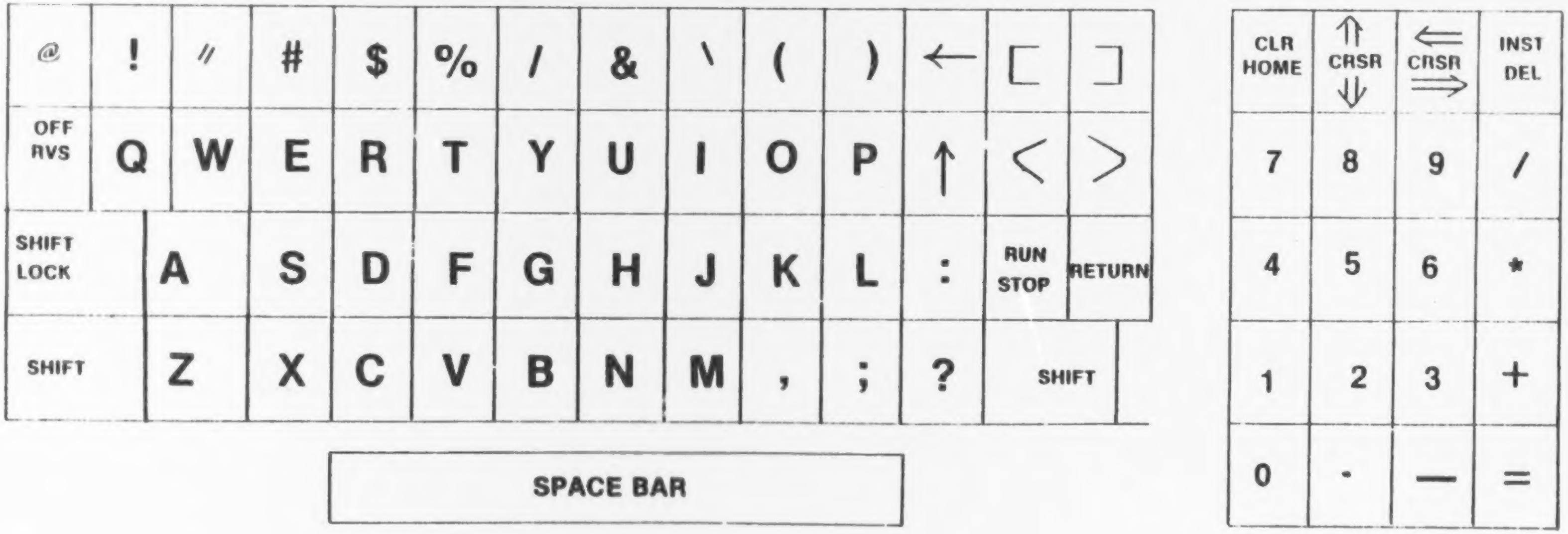
  THESE ARE ALL CAPITAL LETTERS.

- Type the same sentence in reverse letters.

  Make the computer type "?SYNTAX ERROR
  READY. "

- Cursor-Dance.
  - Clear display.
  - Place a line of 30 dots on your screen.
  - Erase last five.
  - Erase five more.
  - Erase entire line.
  - Bring cursor down two lines.
  - Type your first name.
  - Bring cursor down one line.
  - Make cursor move two spaces.
  - Type your last name.
  - Bring cursor to start of next line.
  - Type any three words.
  - Type same three words in reverse.
  - Erase entire last line.
  - Retype it.
  - Type "I am done!"

PET


@ ! " # $ % / & \ ( ) ← [ ]
OFF RVS
Q W E R T Y U I O P ↑ < >
SHIFT LOCK
A S D F G H J K L :
RUN STOP
RETURN
SHIFT
Z X C V B N M , ; ?
SHIFT
SPACE BAR
CLR HOME
⇑ CRSR ⇓
⇐ CRSR ⇒
INST DEL
7 8 9 /
4 5 6 *
1 2 3 +
0 . — =

Lesson 5C

KEYBOARD SKILLS — APPLE II PLUS

PERFORMANCE OBJECTIVES: Students will be able to:

- explain the use of the keyboard, the various keys and special-function keys.
- use these keys and symbols to write programs in the BASIC language.
- demonstrate an understanding of the concept that the keyboard is the primary vehicle through which the user communicates with the CPU.

VOCABULARY: NEW: function, cursor, prompt, space-bar, upper-case, lower-case, shift, control, escape, reset, rept, symbols.

MOTIVATION:

- Elicit responses to the following:
  - How does a driver control a car?
  - How does a pilot control an airplane?
  - How does a captain control a ship?
  - How do you operate a touch-tone telephone?
  - How does an office worker operate a mimeograph machine, adding machine, etc?

AIM: What are the special controls that we use to operate the computer?

DEVELOPMENT:

1. How is our computer similar to a typewriter?
   (Both have keyboards which enable us to operate the device.)
2. What are the differences in the keyboard of a typewriter and a computer keyboard?
   (special symbols and keys, i.e. < , > , 0)

3. What is the little blinking square of light on the screen called? What is its function?
(The little blinking square of light on the screen is called the cursor or prompt and is used to indicate where the next available space is.)

4. Find the space-bar on the computer keyboard. Press it. What happens?
(The cursor on the screen, it moved as you pressed the space bar.)

5. Make the cursor move six times.

6. What happens when you press the right arrow key?
(The cursor moves one space to the right.)

7. What happens when you press the left arrow key?
(The cursor moves back - or to the left.)

8. What are the different functions of these keys?
(The right arrow key advances the cursor one space and the left arrow key moves the cursor back one space.)

9. What have you typed on the screen so far?
(Nothing, but spaces are a form of output.)

10. How do you make letters or numbers appear on the screen?
(You must press the appropriate keys.)

11. How do you clear the screen so that you can start writing at the top of the screen?
(Press the ESC key, then type the symbol @ by pressing the shift key down and pressing the P. You can also type HOME and then press return.)

12. Clear the screen using one of the methods described in #11; and then type your initials. Watch the screen. Where is the cursor now?
(The cursor appears immediately after the last letter on the screen.)

13. Type one more letter and watch the screen. What has happened to the cursor?
(The new letter appears, followed by the cursor.)

14. What does the position of the cursor tell the user?
(The position where the next letter/number/symbol will appear when it is typed.)

15. Make your initials appear five spaces apart from the last letter on the screen.
(Press space bar five times and then type your initials.)

16. Type in HOME or press the ESC key; then press the shift key and the letter P at the same time. What happens?
(The screen is cleared and the cursor is now at the beginning of the first line of the screen.)

17. Now type in your complete name (first and last), leaving one space between names. How might you erase what you just wrote?
(You could use the left arrow key to "back up" and erase the word.)

18. There is an easier way to erase an entire line. Type HOME and press return. What happened?
(The screen is clear but what you've typed is still in the computer's memory.)

19. We have discovered that the computer has special controls that let you do much more than you could with a typewriter. Let's start a list in our notebooks: (Elicit and place following chart on boards.)

KEY(s)	FUNCTION (JOB) PERFORMED
Space bar	Moves cursor ahead; makes blank.
Right arrow	Moves cursor ahead.
Left arrow	Backspaces (Moves cursor back).
ESC + Shift, P	Clears screen
RETURN	Moves cursor down to start of next line. Computer tries to follow instructions.

20. We will be adding to this chart, but let's get some practice using what we have learned. Clear the display.
    - Type "I am learning to control the computer."
    - Erase the last letter.
    - Erase the entire line.

21. Retype the same sentence, but put two spaces between each word. Now continue to tap the space-bar and watch the cursor. What happened when the cursor reached the right-hand edge of the screen? (Cursor jumped down to start of next line.)

22. What other key will make the cursor move down?
(RETURN.)

23. Make the cursor move down almost to the bottom of the screen. Now watch carefully and make it move down to the bottom. Press RETURN again. What is happening on your screen? (The cursor is stationary, but the display is moving off the screen.)
   - Explain: The screen has a fixed number of 24 lines. To add additional lines, the topline is moved upward and is lost. This is called "scrolling."

24. How might you move the cursor upwards? (Hold down ESC and press I.)

25. How might you move the cursor back down? (Hold down ESC and press M.)

26. Now try this: Type "A", ESC M, "B", ESC M, "C", ESC M, and so on. What do you see? (The letters form a diagonal line.)

27. Type HOME and press RETURN. Now hold down the REPT key and press A once. Release the REPT key. What happened? (The letter A repeated until the REPT kep was released.)

28. The combination of REPT and ESC I or M can make the cursor move up or down on the screen. Hold down REPT, press ESC and I. Release. What happened? (The cursor moved down.)

29. Hold down REPT, press EXC and M. Release. What happened? (The cursor moved back up.)

30. The keys ESC and J move the cursor to the left. What other key does this? (Left arrow.)

31. The keys ESC and K move the cursor to the right. What other key does this? (Right arrow.)

32. We've learned a lot about controlling the computer. Let's turn to our chart again.

KEY (S)	FUNCTION (JOB) PERFORMED
REPT	Repeats any letter pressed.
ESC,I	Moves cursor up.
ESC,M	Moves cursor down.
ESC,J	Moves cursor left.
ESC,K	Moves cursor right.

SUMMARY:

- How are computer keyboards similar to typewriter keyboards?

- How are computer key boards different from typewriter keyboards?

- What do we call that group of keys which allows us special control over the display?

- What is the name of the blinking block of light? What is its purpose?

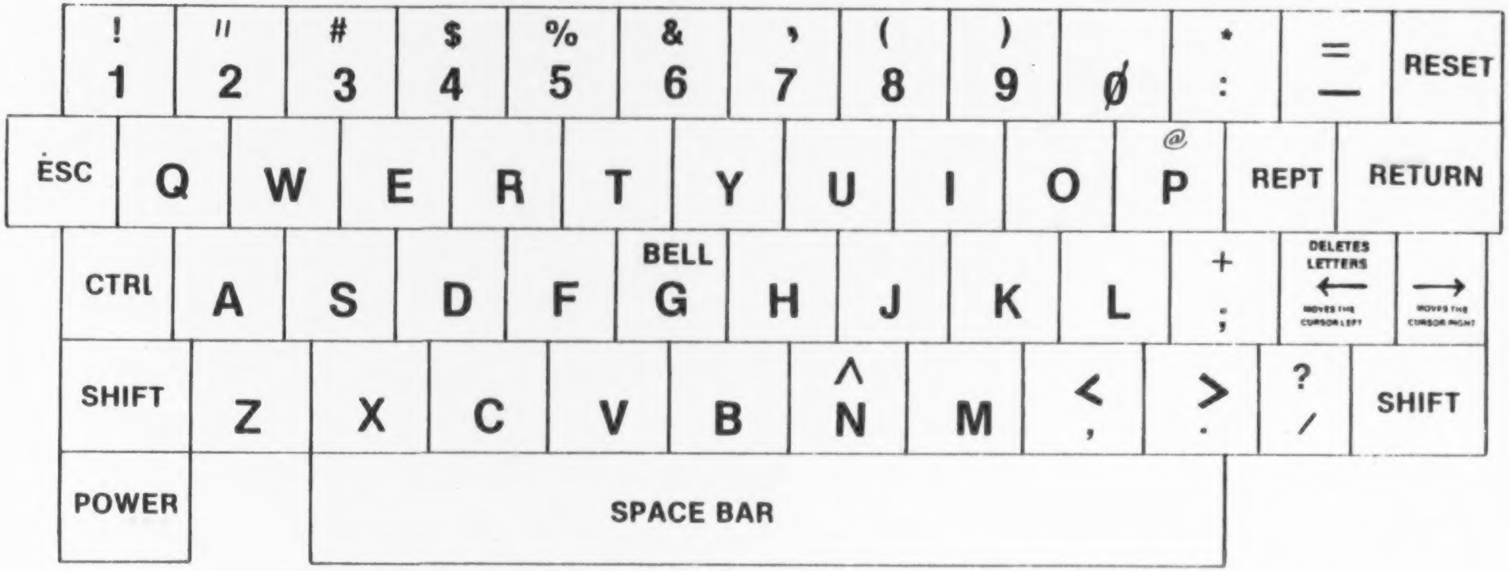
- Describe what happens when a user presses each of the following keys or combination of keys:
    - Space-bar
    - RETURN
    - REPT
    - ESC,I
    - Left arrow

APPLICATIONS:

- Type the numbers 1 - 9 in a diagonal line.

- Type any letter several times by using the REPT key.
    - Cursor dance
        - Clear display.
        - Place a line of 30 dots on your screen.
        - Erase last five.
        - Erase five more.
        - Erase entire line.
        - Bring cursor down two lines.
        - Type your first name.
        - Bring cursor down one line.
        - Make cursor move two spaces.
        - Type your last name.
        - Bring cursor to start of next line.
        - Type any three words.
        - Bring cursor to start of next line.
        - Type "I am done!"

APPLE II PLUS


! 1
" 2
# 3
$ 4
% 5
& 6
' 7
(8
) 9
0
* :
= —
RESET
ESC
Q
W
E
R
T
Y
U
I
O
@ P
REPT
RETURN
CTRL
A
S
D
F
BELL G
H
J
K
L
+ ;
DELETES LETTERS ←
MOVES THE CURSOR LEFT
→
MOVES THE CURSOR RIGHT
SHIFT
Z
X
C
V
B
^ N
M
< ,
> .
? /
SHIFT
POWER
SPACE BAR

Lesson 6

THE PRINT STATEMENT: MESSAGES

PERFORMANCE OBJECTIVES: Students will be able to:

- identify the function of the PRINT statement.
- use the PRINT statement to display a message on the video display.

VOCABULARY: NEW: PRINT, immediate mode, syntax error

REVIEW: cursor, RETURN (ENTER) key, editing keys

MOTIVATION: Create analogy for the class:

- Pretend it's the old days of elementary school. You have just mastered the use of that complex tool, the pencil, and have learned to print.
- What is the skill of printing?
- Why is it an important skill?
- Discuss: It allows a person to get a thought onto paper (abstract to concrete), and it allows others to read it later.
- Computer analogy: Computers can make a message known to the outside world but only when they are properly instructed. We will learn to instruct our computers to PRINT our message.

AIM: How can we convince the computer to display our message on the screen?

DEVELOPMENT:

1. Introduction

What have we used to communicate with the computer? (the keyboard) We will begin to see how we can actually exchange thoughts and converse with the computer.

2. Methods of giving instructions: Programming mode, immediate mode.

   What is the programming mode? (It allows us to give the computer a series of instructions which it keeps in its memory for later performance at our command.)

   What is the immediate mode? (You can give the computer a direct command and see the results right away.)

3. Typing Messages

   How will you introduce yourself to the computer? (Type name.)

   Where on the screen did the characters you typed appear? (Wherever the cursor is located.)

   What key must we press before the computer will understand us and try to do what we tell it to do? (RETURN/ENTER.)

   What did the computer display on the screen after you pressed RETURN (ENTER)? (An error message.)

   What does an error message mean? (It is the computer's way of saying that it doesn't understand, in terms of its own system of grammar, what you typed in.)

4. Giving commands that the computer understands

   If your mother gave you $5.00 but no instructions, and then asked you two minutes later if you were back from your errand, what would you think? (Confusion/Need precise instruction.)

   What did the computer think when you typed your name and pressed RETURN/ENTER? (Confusion/Needed to know what to do with it.)

   What instruction do you think would work? (PRINT.)

   How would we give this instruction to the computer? (Type PRINT and your name, press RETURN/ENTER.)

   What happened when we tried it? (A zero appeared, showing that the computer tried to give the name a numerical value.)

5. Solving the problem

- What should we do to tell the computer to PRINT the name exactly as we type it? (PRINT "NAME" and press RETURN (ENTER): The computer must be told:
  - the message contains words
  - where the message starts and ends
- What happened this time? (The computer understood. It printed the name, without the word PRINT or quotation marks, as we told it to do.)

APPLICATIONS: Instructions:

- For each of the following examples, decide what the computer will do. If the computer would not print what you want, correct the command. Check your answers by typing each example into the computer.
  - PRINT "HELP! I'M BEING HELD CAPTIVE IN THIS BOX."
  - PRINT GOOD MORNING, FRED.
  - PRINT "I LIKE MY COMPUTER CLASS."
  - PRINT WHAT IS TWO PLUS TWO?"
  - "PRINT I'M A COMPUTER. WHAT ARE YOU?"
  - PRINT "I'M YOUR SERVANT; YOUR WISH IS MY COMMAND."

SUMMARY: What steps must we follow to put a message on the screen?

1. Use the command PRINT, spelled correctly, to tell the computer to display our message on the screen.
2. Tell the computer that our message contains words (or a combination of letters and numbers) by enclosing the message in quotes.
3. Tell the computer to perform the task we gave it by pressing RETURN (ENTER).

EXERCISES (ASSIGNMENT):

- Determine what will appear on the screen for each of the following examples. If you believe that any will not work or will produce an error message, please correct it.

- PRINT HI, AM I YOUR FAVORITE COMPUTER?"
- PRINT "MARY LOVES JOHN."
- PRINT "WHAT'S YOUR FAVORITE SUBJECT IN SCHOOL?
- "MY NAME IS SYLVESTER, THE COMPUTER."
- PRIMT "WHAT'S UP, DOC?"

- Use your imagination to make up your own messages. Then write the instructions which will tell the computer to display your messages on the screen.

Lesson 7

THE PRINT STATEMENT: NUMERICAL EXPRESSIONS

PERFORMANCE OBJECTIVES: Students will be able to:

- identify the four arithmetic operation symbols.
- use the PRINT statement to evaluate arithmetic expressions.

VOCABULARY: NEW: operators (+, -, *, /), string, scrolling

REVIEW: PRINT, Clear screen

MOTIVATION: Playing computer:

- Determine the output of each example. Correct any errors.
    - "WELCOME BACK"
    - PRINT "WHAT'S NEW?"
    - PRINT HI THERE!
- What is the output of each example?
- Review: Why do we need the PRINT command? What is the form required with it? (quotation marks)
- What is the collection of characters inside the quotation marks called? (string)

  Example: The word "STRING" is a string of 6 characters.
- Discuss: Another function of the PRINT statement is to do numerical calculations and to display the result.

AIM: How can we make the computer solve numerical problems and tell us the answer?

DEVELOPMENT:

1. What besides words (in the form of "STRINGS") can we type in from the keyboard? (numbers and mathematical symbols)

   - Let's see how the computer reacts when numbers are typed in from the keyboard.

   - Have the students at the computers type:

     3 + 5 (RETURN)

   - What appeared on the screen? (The cursor disappeared when they pressed return and reappeared a short time later with nothing displayed on the screen.)

   - What did the computer do when the enter key was pressed, even though it didn't show it to us? (In this example the computer added the two numbers but is keeping the answer a secret from us.)

2. How we can get the computer to tell us the answer?

   - What word have we already learned that tells the computer to display information on the screen?

   - What happens if we type this line?

     PRINT "3 + 5" (RETURN)

   - Why does "3+5" appear?

   - If words or other symbols are enclosed in quotes, the computer sees them as strings and prints them exactly as they were entered.

3. How must we type in this statement if we want the computer to do the arithmetic, and show us the results? Verify your answer by typing it into the computer.

   - Answer: PRINT 3 + 5 (RETURN)

   - What appeared on the screen? (The computer in this case did the arithmetic and printed the answer, 8.

   - Type: PRINT 10 - 2 (RETURN)

4. What general rules about the PRINT statement can we formulate based upon our observations?

   - The computer will print STRINGS (words, numbers, and symbols in quotes) exactly as entered.
   - Words not in quotes will not be printed.
   - Numbers not in quotes will be acted on mathematically.

5. How could we write a statement to multiply two numbers (for example, 3 times 5) and display the results?

   - Discuss: operator (mathematical symbol) between the two. Why can't we use 'x' as we do in arithmetic? (To the computer each key on the keyboard has a unique function; 'x' can only represent the letter.)
   - What other key isn't used often and was chosen for this function? (the asterisk *). Write the expression '3 times 5' in proper form and type it into the computer.

```
PRINT 3 * 5    (RETURN)
```

6. What numeric operation do we have left? (division).

   - What symbol on the computer keyboard is used to represent division? (the slash)
   - Type: `PRINT 10/5     (RETURN)`

7. Now that we have done these examples, the screen is very cluttered. The information on the top of the screen has been pushed off.

   - The contents of the screen moving upward is called "scrolling."
   - How can we clear the screen so we can try some more examples? (Review the clear screen function: IBM - CTRL/HOME; Pet - CLR/HOME; and TRS-80 - CLEAR; Apple - type the word HOME.)

APPLICATIONS: Determine the output of each example. Correct any errors. Check your results by typing each example into the computer.

- `PRINT 9 * 2`
- `PRINT "9*2"`
- `PRINT "I like you"`
- `PRINT "3 + 5 =" 3+5`
- `PRINT "TEN DIVIDED BY FIVE IS " ;10/5`
- `PRINT "***** HELLO *****"`

SUMMARY: What steps must we follow to have the computer act like a calculator? (Place responses on board.)

- Type the statement PRINT.

- Use the following math operators: + (addition); - (subtraction); * (multiplication); / (division) between the numbers you wish to use.

- Do <u>not</u> place numbers in quotes if you want them to be acted upon mathematically.

EXERCISES (ASSIGNMENT):

1. Determine the output of each example. Correct any errors.

   - `PRINT 5 X 2`
   - `PRINT WHAT'S UP DOC`
   - `12 + 6`
   - `PRINT "DOES 6 * 2 =" 3 * 4`
   - `PRINT 12/6`

2. Make up examples of your own.

Lesson 8

PROGRAMMING MODE

PERFORMANCE OBJECTIVES: Students will be able to:

- use the PRINT command to display output.
- use the comma to make columns on the computer's monitor.
- define and apply the RUN and LIST commands.
- write a simple program using PRINT statements.
- state the difference between the immediate/direct and programming modes.

VOCABULARY: NEW: RUN, LIST, comma, columns, programming mode

REVIEW: PRINT, Immediate/Direct Mode, colon,

MOTIVATION:

- Use the immediate mode to display your name, your address, your city, state, and ZIP code.

AIM: How do we use the PRINT statement?

DEVELOPMENT:

1. Review the use of the immediate/direct mode.
2. Review PRINT statements. Emphasize the need for quotation marks around the information to be displayed.
3. A program is a well organized set of instructions. In the programming mode:
    - The computer stores instructions until directed to put them into operation.
    - Instructions are assigned line numbers to keep them in order while stored in computer memory.
    - The computer follows each set of instructions in the order assigned by the line numbers.

- The last program line should be END. END tells the computer that the program is finished.

4. Have students change the immediate mode program used as the motivation exercise to the programming mode.

   - Discuss the order of statements.

   - Discuss the use of numbers in multiples of ten.

   - Discuss why we press ENTER (RETURN) after each line.

     - Acceptable answer:

```
10  PRINT "your name"
20  PRINT "your address"
30  PRINT "your city, state, ZIP code"
40  END
```

5. Certain commands are necessary to programming:

   - LIST - instructs the computer to display the program in memory.

   - RUN - instructs the computer to execute the program.

   - NEW - erases current program in memory.

6. Let's try an example:

   - Write a program listing the days of the week.

   - Place a comma at the end of the first four PRINT statements. (Discuss the effect of the comma on the output: Commas cause output to print in computer made columns.)

7. Rewrite the day of the week program into a shorter program.

APPLICATIONS: Direct students to write programs that will do the following:

1. List the names of five friends.

2. List the months of the year in columns.

3. List the names and ages of three people. (Names in one column, ages in another column.)

SUMMARY:

- Review the use of the LIST, RUN, and PRINT commands.
- Review the use of the comma.
- Compare the immediate/direct mode with the programming mode.

EXERCISES: Write the programs that will do the following:

1. Make a chart showing the names and addresses of three people.
2. Make a list of five actors.
3. Make a chart showing your five favorite records along with their recording artists. Use separate columns for records and artists.

Lesson 9

PROGRAM MODIFICATION

PERFORMANCE OBJECTIVES: Students will be able to:

- explain the significance of line numbers.
- explain the effect of using duplicate line numbers.
- delete program lines.
- edit program lines.
- create blank lines.
- demonstrate use of LIST and RUN commands.

VOCABULARY: NEW: insert, edit, delete

REVIEW: LIST, RUN, line number

MOTIVATION: Place the following program on the board exactly as shown:

```
60 PRINT "COMPUTERS"
31 PRINT "DON'T"
55 PRINT "LIKE"
 8 PRINT "PEOPLE"
20 PRINT "WHO"
77 PRINT "ARE"
85 PRINT "SILLY"
```

- What might be the output of this program? (Some adept students may immediately recognize the "trick." Ask them to not spoil the surprise.)
- Have students clear computer memory and then type in the above program.
- We've just typed a program into memory. How do we make it work? (We must type RUN.)
- RUN your program and tell us what appeared on screen. (Output: PEOPLE WHO DON'T LIKE COMPUTERS ARE SILLY one word per line.)

- Why did we get this message instead of COMPUTERS DON'T LIKE PEOPLE WHO ARE SILLY? (Line numbers determine order.)

- How did the line numbers affect the output? (Computer executed commands according to line number order from lowest to highest, despite order in which they were typed.)

- Since line numbers are so important to the computer, what might be the aim of this lesson? (Elicit AIM.)

AIM: How can we use line numbers to change a program and its output?

DEVELOPMENT:

1. Imagine that you were surprised by the output of our program. You decide to check the program for errors. How could you instruct the computer to put your program on the screen? (Type LIST.)

    - Type LIST and examine the program.

    - Type RUN and compare the output to the LISTed program.

    - What did the computer do to the program we originally typed in from the chalkboard? (placed lines in proper numerical sequence in order to correctly execute the program.)

2. What would we do to make the output read PEOPLE WHO DON'T LIKE TEACHERS ARE SILLY? (We need to change "COMPUTERS" to "TEACHERS" in line 60.)

    - How do we change a line in a program? (Retype line number: Type new statement.)

    - In line 60 change "COMPUTERS" to "TEACHERS." (Students will type: 60 PRINT "TEACHERS.")

    - How do we verify that our output will now be different? (Type RUN.)

    - Please RUN your revised program.

3. What is the difference between RUN and LIST? (RUN causes execution of the program. LIST causes program statements, or Source Code, to appear on the screen.)

- How would a programmer use these two commands to correct a program? (Type LIST, make changes, type RUN to verify.)

4. Let's make some more changes in our program. How might you instruct the computer to print blank spaces between lines of the output? (Use "blank" PRINT; i.e., line number followed by PRINT.)

   - Revise your program to make the output show a blank line after each of the first three lines.

     Acceptable revision:

     ```
      8 PRINT "PEOPLE"
     10 PRINT
     20 PRINT "WHO"
     25 PRINT
     31 PRINT "DON'T"
     35 PRINT
     ```

   - RUN your program to verify that it was changed.

5. In the example above, why did we have to create new line numbers? (To avoid destroying an existing line.)

   - Prove this to yourself. LIST your program. Type:
     8 PRINT.
     RUN your program and describe the output.
     (output should be WHO DON'T LIKE TEACHERS ARE SILLY.)

   - Why did the word "PEOPLE" disappear?
     (8 PRINT "PEOPLE" was replaced by 8 PRINT.)

6. Restore the word "PEOPLE" to your output in its proper place. (Students will LIST and type: 8 PRINT "PEOPLE.")

   - How would you make the output read RED-HAIRED PEOPLE WHO DON'T LIKE TEACHERS ARE SILLY? (You must instruct the computer to add "RED-HAIRED" by adding a new line, numbered lower than 8; i.e., 5 PRINT "RED-HAIRED.")

   - How would you make the output read RED-HEADED PEOPLE ON MARS WHO DON'T LIKE TEACHERS ARE SILLY? (You must type: 9 PRINT "ON MARS" as 8 and 10 are already used.)

   - Make these two changes and RUN your program. (New output should be RED-HAIRED PEOPLE ON MARS...etc.)

7. Another term for making changes such as these is "program modification." The modifications include adding, dropping, or changing lines. Let's learn some technical words.

   - When we add a new line anyplace in the program we are said to be "inserting" a line. How do we "insert"? (Depending on whether we wish to add before or after an existing line, we must use a line number with proper numerical relation to existing line numbers.)

   - Another term for removing a line is to "delete." How do we "delete" an existing line? (Type proper line number and retype line, making desired changes.)

   - The correct term for changing part or all of a line is to "edit." How do we "edit" an existing line? (Type proper line number and retype line, making desired changes.)

   - How does a programmer verify that the computer actually made the proper modifications? (Type RUN to check output.)

8. A trick of programmers is to use line numbers in multiples of either 5 or 10, such as 10...20...30...40...etc. Why is this an example of excellent programming technique? (Allows for an easy insertion of new lines.)

   - What might happen if a programmer numbered his or her lines simply 1...2...3...4...5...etc.? (It would be impossible to insert new lines.)

APPLICATION:

- Have students type NEW and then type the following 2-line program:

```
30 PRINT "HELLO!"
50 PRINT "HOW ARE YOU?"
```

- Students will then make the following modifications to this program:

  - Insert a blank line between HELLO and HOW ARE YOU?

  - Insert two new lines: the first should read I AM A and the second should read COMPUTER. Both will appear below the existing output.

- Edit line 30 to read "HI!!!"instead of "HELLO!"

- Delete line 50 from output.
- Insert correct brand name of computer above word "COMPUTER."

- Insert a line of 20 asterisks (*) above the first and below the last line of output.

N.B.: Each modification should be verified by typing RUN before proceeding to the next modification.

SUMMARY:

- What are the technical terms for the three kinds of program modifications about which we learned?

- Explain the meaning of each of the three terms.

- Explain the difference between LIST and RUN.

- Explain the reason for numbering program lines in multiples of 5 or 10.

EXERCISES (ASSIGNMENTS):

- Write a simple two-line program using PRINT statements. Then:

  - Using ordinary English, describe a modification that you might wish to make to this program. Your modification could be to edit, to insert, or to delete. Be sure to describe adequately the modification in your own words.

  - Below the description, write the actual programming statements, complete with correct line numbers, which would accomplish the modification described.

  - Repeat the last two steps for a different type of modification. (If you edited for the first modification then insert or delete for the second.)

Lesson 10

NUMERIC VARIABLES AND ASSIGNMENT STATEMENTS

PERFORMANCE OBJECTIVES: Students will be able to:

- explain how the computer stores information.
- describe how variables are used in programming.
- define and write numeric variables.
- describe and use assignment statements with numeric variables.

VOCABULARY: NEW: memory locations, variable, numeric variables, assignment statement

MOTIVATION: Code games:

A = 1, B = 2, C = 3, D = 4, E = 5,
F = 6, G = 7, H = 8, I = 9, J = 10

. Using the above code, rewrite this message:

6135 18314

. Using the above code, rewrite the following examples.

. B * E
. H / D
. G - C
. A + J
. D + (F * I)

. What was the purpose of these games? (To see how letters can represent numbers and vice versa.)

AIM: How do we represent numeric data in the computer?

DEVELOPMENT:

1. What is the process the computer follows to store information?

   - . Information is converted into a code that the computer can read.
   - . Information is then stored in memory locations inside the computer.
   - . Each program line is given a location in memory.
   - . The computer remembers where the information and lines for each program are stored.

2. Compare computer memory locations with teachers' mailboxes in the school office.

   - . Emphasize that there can be only one piece of information in a memory location at any given time.
   - . Analogy: Each teacher receives only one letter at a time.

3. How were codes used in the motivation exercise? (Each letter was given a number value. Each number was given a letter value.)

4. Variables: For computers, a symbol used to represent a number or expression is called a variable.

   - . Variables are like the names on mailboxes.
   - . We use variables to name areas inside the computer called memory locations where data is stored.

5. Numeric variables name memory locations for numbers or arithmetic expressions. In BASIC, a numeric variable can be:

     - . a single letter of the alphabet (A - Z) or
     - . a letter followed by another letter or digit (0 - 9).

   - . Many computers only recognize the first two characters. Therefore, NUM1 and NUM2 would be treated as the same memory location.

- Since a memory location can hold only one piece of information, assigning the same variable name to two pieces of information will erase the first one from the memory.

- What are some examples of numeric variables? N, DC, AGE, A3

- Write several variables that you think will work.

6. What was the procedure you used to complete the motivation exercise?

    - How did you know the number each letter represented?

7. How could we tell the computer where a piece of information is stored? (Assign a variable to represent a number.)

    - Let's look at the examples we used at the beginning of this lesson: A = 1, B = 2, etc.

    - How would we write these in program lines?

        - 10 LET A = 1

        - 20 LET B = 2

    - What is the word LET called? (Assignment statement.)

    - What is its purpose? (Sends information to memory location named by the variable.)

8. How many memory locations are used in the program line: 10 LET A = 1? (Two.)

    - The computer stores 10 LET A = 1 in one memory location, then stores the value 1 in the memory location named A.

    - Remember, each program line or instruction is given a location and each piece of information is given a location. The two locations are different.

9. In this program, whenever we use the variable A, what number are we actually using? (The number one.) The assignment statement LET provides the variable with its value.

    - Note: The variable always appears on the left side of the assignment statement.

Example: 10 LET A = 1

```
10 LET A = 1
20 PRINT A
30 END
```

10. In the above program what will appear on the monitor? (The number 1.)

    - Note: Variables used in PRINT statements are not surrounded by quotation marks.

11. What other forms can an assignment statement have?

    - LET A = B (A names the location containing the value of B.)

    - LET A = B + C (A names the location containing the value of B plus the value of C. Both B and C must be given values before this can be used.)

    - LET A = B + 1 (A names the location containing the value of B plus 1. This function can be used to count items.)

12. Let's try to figure out what will appear on the screen if this program is used:

```
10 LET B = 2
20 LET C = 3
30 LET A = B + C
40 PRINT "A="; A
50 END
```

   - The computer will print A=5.

   - Let's run this program on the computer.

APPLICATIONS:

Use the motivation exercise to complete the following:

- Rewrite the code as assignment statements.

- Write a program which would print the values of C, D, F, H and J.

- Modify the program to write output in the following form: A = 1

100

Use the following list to do the exercises below:

A = 4, B = 7, C = 5, D = 10,
F = A + B, G = D/C, H = B * C

- Write a program which will print the values of B, G, and H.
- Write a progam which will print the sum of B, G, and H.

SUMMARY: Review the following:

- What is the procedure the computer follows to store programs and data?
- What are the characteristics of numeric variables?
- How many characters of a variable name does a computer recognize? What are the forms and uses of the assignment statement?

EXERCISES:

1. Explain how the computer stores the following program:

```
10 LET K = 2
20 LET M = 5
30 PRINT K
40 PRINT M
50 END
```

   - Add a line to have the computer print the sum of the two numbers.

2. Write the output of the following program:

```
10 LET N = 12
20 LET G = 4
30 PRINT N; "+"; G; "=";
40 PRINT N + G
50 END
```

   - Add two lines to have the computer print the result when G is subtracted from N.

3. Given A = 5, B = 7, C = 9, write a program which will calculate the following: 7 x 9, 9 - 5, 7 + 5 and print the results in sentence form.

BEST COPY AVAILABLE

4. Pencils cost 15¢ each, notebooks cost $2.95 and a package of paper costs $1.50. Write a program which will compute the cost of:

   - a pencil and a notebook.
   - a notebook and a package of paper.
   - 5 pencils, 2 notebooks and 3 packages of paper.

   - Rewrite the programs to print the results in sentence form.

Lesson 11

STRING VARIABLES AND ASSIGNMENT STATEMENTS

PERFORMANCE OBJECTIVES: Students will be able to:

- explain how variables are used in programming.
- define and write string variables.
- describe and use assignment statements with string variables.
- use string variables without the command LET.
- make the computer add strings.

VOCABULARY:	NEW:	alphanumeric characters
	REVIEW:	memory location, string, assignment statement

MOTIVATION: Review numeric variables:

- Given A = 16, B = 4, C = 8, write a program which will calculate D if D = (A/B) + C. The output should be in sentence form.
- Why should the output be in sentence form?

AIM: How does the computer handle words and combinations of numbers and words?

DEVELOPMENT:

1. How does a computer use numeric variables and assignment statements?
   - For the program you wrote a few minutes ago, how did the computer store your program lines and data? (They are stored separately in computer memory.)
   - Let's devise a chart to show memory locations and the assignment variables of each one used.

2. How is the following information different from what we've used before?

   Boris Baker
   55 Park Avenue
   New York, NY 10022
   469-4361

   - Since these items contain letters, numbers, and other keyboard characters (including spaces), what kind of data could we call them? (Alphanumeric data.)

3. Let's review simple PRINT statements:

   - How did we print our names? (PRINT "name")

   - What do we call anything within quotation marks? (String.)

   - What is the definition of a string? (A string is a series of alphanumeric characters.)

   - Why can't we use a string in calculations? (The computer can't give a numeric value to a string.)

   - Besides our names, what are some examples of strings? (HELLO, #5A, 926XJR.)

4. How do we tell the computer to remember the string? (We use an assignment statement.)

   - What is this assignment statement called? (String variable.)

   - What form do we use? (A string variable begins with a letter and ends with a dollar sign, $.)

   - How many letters can we use before the dollar sign? (Most computers recognize only the first two characters of a variable name.)

   - What are some examples of string variables? (A$, N$, AN$, AI$.)

   - Note: A telephone number is alphanumeric because of the hypen (-) used to separate the digits.

5. What is the function of an assignment statement? (An assignment statement is used to send data to the memory location named by the variable.)

- How do we use quotation marks with a string variable? (Data being stored by a string variable must be enclosed)

- What are some examples? (A$ = "HELLO," etc.)

- How could we write a program to print HELLO?

```
10 LET A$ = "HELLO"
20 PRINT A$
```

6. How would we write a program to say hello to us by name?

```
10 LET A$ = "HELLO"
20 LET B$ = "NAME"
30 PRINT A$
40 PRINT B$
50 END
```

7. Trick #1

- What word tells us that an assignment statement is being used? (LET.)

- Let's try a shortcut and type in the program we just did without the word LET in lines 10 and 20.

- What happened? (The program worked exactly the same way as before.)

- Why? (The computer recognizes variable names, either numeric or string, without the word LET.)

8. Trick #2

- Write a program to print GOODBYE using GOOD and BYE as separate words.

```
10 A$ = "GOOD"
20 B$ = "BYE"
30 PRINT A$
40 PRINT B$
50 END
```

```
10 A$ = "GOOD"
20 B$ = "BYE"
30 C$ = A$ + B$
40 PRINT C$
50 END
```

- How could we get the computer to add the two strings for us and print them on the same line?

```
10 A$ = "GOOD"
20 B$ = "BYE"
30 PRINT A$ + B$
40 END
```

APPLICATIONS:

- Write a program to print the word sunshine using sun and shine as separate words.
- Make a chart to show the way the computer stores the program.
- Find two other ways to accomplish the same thing.

SUMMARY:

- How does the computer store programs and data?
- What are alphanumeric data and strings?
- What are the characteristics of string variables?
- Compare string and numeric variables. (Numeric variables accept only numeric data. String variables accept all alphanumeric data. Numbers stored as strings cannot be used in calculations.)
- What are the forms and uses of assignment statements?
- What tricks did we use?

EXERCISES (ASSIGNMENT):

1. Given A$ = "TO," B$ = "HER," C$ = "GET," write a program which prints TOGETHER.
2. Write a program in which the output looks as follows:

```
HELLO
STUDENT
GOODBYE
COMPUTER
```

3. Rewrite the above program to make the computer add GOOD and BYE to get the same output.
4. Write a program to print your name, address, city and state, and telephone number.

Lesson 12

UNCONDITIONAL BRANCHING (GOTO)
PRINT FORMAT — COMMA AND SEMICOLON

PERFORMANCE OBJECTIVES: Students will be able to:

- describe the output resulting from the use of commas in a PRINT statement.
- explain the concept of unconditional branching.
- use the GOTO command in programming.
- describe the output resulting from the use of semicolons in a PRINT statement.
- use semicolons in program lines.

VOCABULARY:	NEW:	print format (in reference to the use of the comma and semicolon), zone, GOTO)
	REVIEW:	PRINT

MOTIVATION: Let's all go on a big ego trip and fill the computer's screen with our names.

- What is the first program line we would use? (10 PRINT "NAME")
- What output would we get from this instruction? (The computer will print the name once on the screen.)
- If we wanted to print our names again below this, what could we do? (We could add line 20 PRINT "NAME")
- If we wanted to print a whole column with our names, what shortcut could we use? (Tell computer to do the same thing over and over again.)
- What program statement would do this? (GOTO)
- How would the program look?

```
10 PRINT "NAME"
20 GOTO 10
```

- How does this work? (Explain the concept of repeated action.)
- How do we stop the program from running? (Explain BREAK or RUN/STOP key.)
- Why doesn't this accomplish what we wanted to do? (There is only one column displaying the name, and not the full screen.)
- What do we want to find out? (Elicit AIM.)

AIM: How can we "format" the computer's output on the screen with very simple program statements?

DEVELOPMENT:

1. What is one way we could get the computer to print our names on the same line as it did already? (10 PRINT "NAME" "NAME" "NAME" etc.)
   - What punctuation have we learned that might make this easier? (COMMA.)
   - What would the program be?

     ```
     10 PRINT "NAME",
     20 GOTO 10
     ```

   - What would the output look like? (The screen would have two columns with our names.)
   - How could we get four columns?

     ```
     10 PRINT "NAME", "NAME", "NAME", "NAME".
     20 GOTO 10
     ```

2. What other punctuation mark might be even better? (Semicolon.)
   - Type the following program and run it:

     ```
     10 PRINT "NAME";
     20 GOTO 10
     ```

   - What does the screen look like now? (The screen is filled with your name.)
   - How could we make the screen easier to read? Add a space between the many appearances of the word.)

- What must we do in order to separate the word(s)? (Insert a space before we close the quote: 10 PRINT "NAME ";)

- What can we say about the role of the semicolon at the end of program line 10? (It makes the computer print the next item on the same line.)

- Now let's see what happens if, instead of printing a word (string), we wanted to print a number.

3. Let's substitute a new line 10, this time containing a number. Type and run the new program.

```
10 PRINT 5;
```

- What does the screen look like? (On most though not all computers, the 5s are separated by a space.)

- Why do you think this happened, since before, when we printed a word, we had to manually insert a space? (Most computers leave a space before numbers for a positive or negative sign; if it is positive, the computer doesn't print the + sign, and a space appears instead.)

4. Now let's change our program by returning to columns. Write a program to put the word COMPUTER in columns.

```
10 PRINT "COMPUTER",
20 GOTO 10
```

- What does the screen look like now? (The word COMPUTER is in columns on the screen.)

- What can we conclude about the use of the comma at the end of a PRINT statement? (A comma makes the computer jump to a new "zone." The columns are called print zones.) These zones are predetermined by the computer, depending on the number of characters that can be typed on a screen and differ from computer to computer.)

- How many zones does your computer print in?

5. Now let's try a little experiment. Type in this little program.

```
10 PRINT "COMPUTER", "COMPUTER"
20 PRINT 5,5
```

- What do you think the output will be?

- Run the program.

- Did the word and number print in the same zone? (yes)

- Why didn't the word COMPUTER and the number 5 start in the same row? (Most computers leave a space for the + sign.)

6. What do you think the screen would look like if we typed in the following program?

```
10 PRINT "GOOD MORNING ";
20 PRINT "HOW ARE YOU?"
30 END
```

- Why must we leave the space after the word "MORNING" in line 10?

- What should we do if we want our program to print the message "GOOD MORNING GEORGE. HOW ARE YOU?" (Add this line: 15 PRINT "GEORGE. ";)

7. What do you think will happen when we use comma, and semicolons with numeric or string variables?

- Predict the outcome of the following program, which uses string variables and the semicolon:

```
10 A$ = "SUPERCALI"
20 B$ = "FRAGILISTIC"
30 C$ = "EXPIALIDOCIOUS"
40 PRINT A$;B$;C$
```

- Run the program.

APPLICATION:

Predict the outcome of the following programs:

```
10 PRINT "THE TIME IS ";
20 PRINT 3;
30 PRINT" O'CLOCK"
```

(Notice the space to keep the 3 from bumping into the O'CLOCK.)

```
10 PRINT "I'M GREAT",
20 PRINT
30 GOTO 10
```

(The comma and the PRINT statement cancel each other out.)

```
10 PRINT "MARY ";
20 PRINT "LOVES "; ";A,B,C
30 PRINT "JOHN"
40 PRINT
50 GOTO 10
```

```
10 PRINT "ZONE 1",, "ZONE3"
20 END
RUN
(Show use of comma.)
```

```
10 PRINT "THE SUM OF 2 AND 3 IS", 2+3
```

SUMMARY:

- What is the difference in screen format between using a semicolon and a comma in a program?
  - When should we use a semicolon in a program?
  - When should we use a comma in a program?
- How do we tell the computer to repeat an action over and over again?

EXERCISES (ASSIGNMENT):

1. Have you ever heard the expression, "It is raining CATS and DOGS"? Write a program that fills the computer's screen with a veritable downpour of "CATS" and "DOGS".

2. Joe, Nancy, and Edna have taken two tests each. Write a program that will give the following output.

```
JOE NANCY EDNA

65   90    73
80   81    75
```

After the third test, Joe wanted to see all of his marks and his average. To get the average, tell the computer to add all of his marks. (He got 75 on test #3.) and divide by 3. Write a program to display his marks and his average. Use sentences in the output as follows:

```
JOE'S MARKS ARE 65, 80, 75

JOE'S AVERAGE IS ...
```

Lesson 13

INTRODUCTION TO FLOWCHARTING

PERFORMANCE OBJECTIVES: Students will be able to:

- identify three flowchart symbols: oval, rectangle, and parallelogram.
- place steps in proper sequence given a list of steps to solve a problem.
- write a flowchart to solve a given verbal problem.
- write a sentence to describe the problem solved by a given flowchart.
- use a flowchart to help write a program.

VOCABULARY: NEW: flowchart, stepwise refinement

REVIEW: oval, rectangle, parallelogram, mean, average, input, output, processing.

MOTIVATION: Sequencing:

- Put the following directions in the correct order:
  1. Close mailbox
  2. Write address on envelope
  3. Walk to mailbox
  4. Put postage on envelope
  5. Open mailbox
  6. Insert envelope into mailbox
- Why do we follow this sequence?
- What problem does this sequence solve?
- How do we solve it?
  - Identify the problem

- Analyze each step of the problem. (This is called stepwise refinement.)
- Break down problem into steps
- How does this apply to writing a program?
- What technique do we use? (Flowcharting.)

AIM: How do we use flowcharts to solve problems?

DEVELOPMENT:

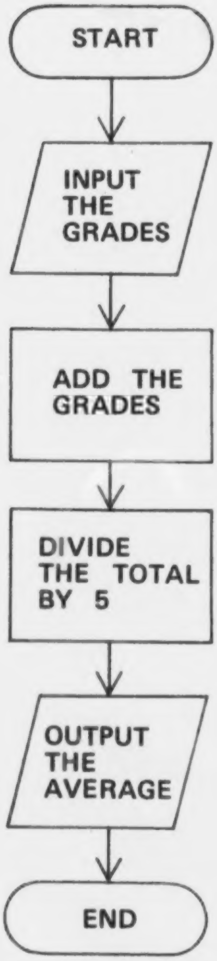
1. You have five major subjects and want to find out the average of your grades on your last report card (English, Social Studies, Science, Mathematics, Language).

    - What steps would you use?
        - supply five grades
        - add the five grades
        - divide by five
        - state the average

2. Let's look at flowchart symbols so we can create a flowchart for this problem.

    - What shape do we use to start or end a flowchart? (Oval.)
    - What shape do we use for information we must supply? (Parallelogram.)
    - What shape do we use for the processing of our information? (Rectangle.)
    - What shape do we use for the solution? (Parallelogram.)
    - What symbol do we use to show the flow from one step to the next? (Arrow.)

3. Let's insert our steps for this problem into flowchart symbols:


START
INPUT THE GRADES
ADD THE GRADES
DIVIDE THE TOTAL BY 5
OUTPUT THE AVERAGE
END

4. Let's review how the symbols were used:

    - Terminals - oval (indicates the beginning or end of a computer program)
    - Input/Output (I/O) - parallelogram (indicates information is coming in or going out)
    - Processing - rectangle (indicates computer is acting upon the information given to it)
    - Lines of flow - arrow (indicates order of steps in proper sequence)

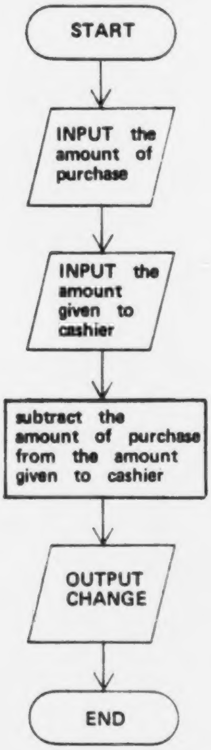
5. What is the purpose of a flowchart?

    - We use flowcharts to solve a problem by arranging the necessary steps in a logical sequence.
    - Flowcharting symbols have been standardized to reflect major components of any computer system: input, processing, output.

6. Let's look at another problem:

    You have a $5.00 bill. You go to the store and spend a total of $4.37. You want to know how much change you'll get.

    - List the steps needed to solve this problem.
        - State the amount spent.
        - State the amount given to cashier.
        - Subtract amount spent from the amount given to cashier.
        - State the amount of change.
    - Create a flowchart for this problem.


START
INPUT the amount of purchase
INPUT the amount given to cashier
subtract the amount of purchase from the amount given to cashier
OUTPUT CHANGE
END

. What purpose did the flowchart serve?

APPLICATION:

- Use the flowchart we just created to help you write a program that will solve the problem.

```
10 A = 4.37
20 B = 5.00
30 C = B - A
40 PRINT C
```

- Rewrite this program to display the following output:

```
I HAD 5.00
I SPENT 4.37
MY CHANGE IS .63
```

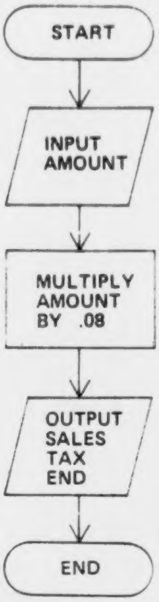
```
10 A = 4.37
20 B = 5.00
30 C = B - A
40 PRINT "I HAD"; B
50 PRINT "I SPENT"; A
60 PRINT "MY CHANGE IS"; C
```

SUMMARY:

- What is the purpose of a flowchart?
- What symbols are used in a flowchart?
- What do the symbols represent?
- How do we use a flowchart to write a program?

EXERCISES (ASSIGNMENT):

- Flowchart the following problems:
  - How to make a peanut butter and jelly sandwich.
  - How to change a light bulb.
  - What you need to do before you go to school.
- Explain in one sentence what problem is solved with the following flowchart:


START
INPUT AMOUNT
MULTIPLY AMOUNT BY .08
OUTPUT SALES TAX END
END

- Write a flowchart and program with sentences for the following problem:

- A newspaper boy earns $3.00 an hour and works 7 hours a week. What is his total salary?

Lesson 14

ORDER OF OPERATIONS IN DIRECT (IMMEDIATE) MODE

PERFORMANCE OBJECTIVES: Students will be able to:

- state the order of arithmetic operations.
- evaluate arithmetic expressions (one or more operations).
- evaluate expressions containing parentheses.
- demonstrate use of the PRINT command.
- demonstrate use of the computer in the DIRECT MODE to calculate values of arithmetic expressions.
- write their own examples to demonstrate the use of the computer as a calculator.
- state the effect of quotation marks in a PRINT statement to determine the output.

VOCABULARY: REVIEW: direct (immediate) mode, PRINT command

NEW: order of operations, parentheses as a grouping symbol, quotation marks

MOTIVATION (DO NOW):

- Do the following examples:
  - 2 + 5 x 3
  - 2 x 5 + 3
  - 2 x 5 - 3
  - 8 x 4 ÷ 2
  - 64 ÷ 8 ÷ 2
- When we do problems like these, what is the order of arithmetic operations?
  - Multiply
  - Divide
  - Add
  - Subtract

- How do we remember this order? (Mnemonic: My Dear Aunt Sally.)
- How is this order related to doing arithmetic problems on a computer?
- How may the computer be used to calculate answers to arithmetic problems? (We use the direct or immediate mode.)

AIM: How do we use the computer as a calculator?

DEVELOPMENT:

1. Type 2 * 5 + 3 on the computer keyboard. Press RETURN (ENTER).
   - What was the output? (Nothing.)
   - Why? (No ouput appears since the PRINT command was not given.)
   - Type the same example but add the PRINT command.
2. Type in two more examples that you create. Include the PRINT command.
3. In the original example, which operation was done first?
   - Review the symbols +, -, /, and * and the order of these operations.
4. Type PRINT "2*5+3".
   - What is the effect of quotation marks?
   - What would appear on the screen from the direct command PRINT "ABCD"?
5. Evaluate each expression using paper and pencil. Then use the computer in the direct mode to check your results:
   - 42 + 44
   - 3 * 33
   - 10 * 8 - 7 * 7
   - 3 + 4 - 6
   - 5 * 4 + 3

        - 4 + 3 * 5
        - 28/4
        - 30 + 6/6

6. How would we solve the following?

    - (4 + 3) * 5 (Add the numbers within parentheses. Then complete the other arithmetic function.)

    - Why do we do the arithmetic on the numbers within parentheses first? Order of operations -- parenthesis is first.)

    - Do the following examples:

        - 4 * 7 - 3
        - 4 * (7 - 3)
        - (4 * 7) - 3
        - (4 + 12)/2
        - 4 + (12/2)

7. Do the following examples using paper and pencil. Try to predict the computer's output when these same problems are done in the direct mode.

    - 4 + 2 * 6/3
    - (4 + 2) * 6/3
    - 4 + (2 * 6)/3
    - 4 + 2 * (6/3)

8. What are we asked to do in this problem: 2↑3

    - What does exponentiation mean? (Using exponents is raising the first number to the power stated by the second number.)

    - How does exponentiation look as an arithmetic problem? (Power number is above and smaller than number to be raised: 2^3.

    - In the order of arithmetic operations, where are exponents? (They are second -- after parentheses.)

    - How can we expand our mnemonic to include parentheses and exponents? (Please Excuse My Dear Aunt Sally.)

- Let's ask the computer to do exponentiation for us. Type in as PRINT statements:
    - 5↑5
    - 8↑6
    - 2↑5

EXERCISES (ASSIGNMENT):

- Do the following examples. Be prepared to check your results using the computer in the direct mode.
    - 20 - 6
    - 4 * 3
    - 2 + 3 * 4
    - (2 + 3) * 6
    - (44 - 7 * 4)/2
    - 70 - 3 * (4 + 2)
    - 14/(3 + 4)
    - 7 + 8/2 - (3 * 2)
    - 12↑2
    - 5 + 3↑2
    - 3↑2 + 4↑2
- Determine the output of each statement:
    - PRINT "A="
    - PRINT 2 * 7 + 5
    - PRINT "2 * 7 + 5 ="; 2 * 7 + 5
    - PRINT "5 * (6 + 7)"
    - PRINT 5 * (6 + 7)
    - PRINT "YOUR SALARY IS"; 5 * 7

Lesson 15

ORDER OF OPERATIONS IN PROGRAMMING MODE

PERFORMANCE OBJECTIVES: Students will be able to:

- distinguish between algebraic and BASIC statements.
- determine the value assigned to a variable by a given LET statement.
- transform a given algebraic statement to its BASIC equivalent.
- determine the output at a specified line of a simple program.
- create an algorithm, a flowchart, and a computer program to solve a problem.

VOCABULARY: NEW: algorithm

REVIEW: exponentiation, variable, LET, order of operations, program, line number, operators, output, PRINT

MOTIVATION:

- What is the output?

```
10 LET Z = 15
20 LET X = 5
30 LET Y = 2 * Z/X
40 PRINT Y
```

- Change line 30 to read 30 LET Y = (2 * Z)/X.
- How is the output different?
- Why is it different?

AIM: How are operations related to programming statements?

DEVELOPMENT:

1. Elicit the following table:

OPERATION	ALGEBRAIC OPERATOR	BASIC OPERATOR	ALGEBRAIC STATEMENT	BASIC STATEMENT
Addition	+	+	A = B + C	10 A = B + C
Subtraction	-	-	A = B - C	10 A = B -C
Multiplication	x	*	A = B x C	10 A = B * C
Division	÷	/	A = B ÷ C	10 A = B/C
Exponentiation	A^2	↑	$B = C^2$	10 B = C 2
Parentheses	( )	( )	A = D(B + C)	10 A = D * (B + C)

2. What is the order of operation?

   - What mnemonic device helps us to remember?

3. Write each of the following algebraic statements as equivalent BASIC statments.

   - $D = RT$
   - $Y = X^2 - 3X + 7$
   - $M = \frac{A - B}{C + D}$
   - $Y = Z(3X - 5)$
   - $A = \frac{1}{2}BH$
   - $F = \frac{9}{5} C + 32$
   - $X = 5N + 10D + 25G$

4. What is the value of R in each BASIC statment? Use X = 10, Y = 15 and Z = 2.

   - LET R = 3 * (X + Y)
   - LET R = Z↑3
   - LET R = (X + Y + 1)/Z
   - LET R = Z↑2 + X↑2
   - LET R = (Z + X)↑2

5. Let's see how the things we've learned can help us solve a word problem.

   - You are 5 years older than your sister. She is twice as old as your cousin. You want to know your cousin's age.
     - Explain in words the steps you would take to get the answer.

- This series of steps (in words) that will solve a problem is called an algorithm.

- From this algorithm, create a flowchart.

- From the flowchart, create a computer program.

APPLICATIONS:

- Your school is having a show. Tickets cost $4.00 for adults and $2.00 for students. You want to know how much the school will earn if 100 adults and 300 students attend.

  - Write an algorithm for this problem.

  - Write a flowchart from the algorithm.

  - Write a computer program from the flowchart. Use sentences in your PRINT lines.

SUMMARY:

- What is the relationship among algorithms, flowcharts, and computer programs?

- Elicit the following chart:

PROBLEM IN WORDS

↓

ALGORITHM OF THE PROBLEM TO BE SOLVED

↓

FLOWCHART (Graphic Symbols, Human Logic and Language)

↓

COMPUTER PROGRAM (Coded Symbols, Computer Logic and Lanuage)

↓

COMPUTER SOLUTION TO THE PROBLEM

EXERCISES (ASSIGNMENT):

- Name the operation which will be carried out first in each of the following statements:
    - 18 - 3 * 2
    - 9 + 11 - 10
    - 5 * (3 + 8)
    - 3 * (2 + 5) - 9/3
    - (5 + 7)/ (8 - 6)
    - 4 + 2 * 3↑2
    - 7 * (3 + 5)↑2
- Rewrite the following problems as BASIC statments.
    - $X = 2 \frac{(A + B)}{C^2}$
    - $R = \frac{A^2 + B^2}{A + B}$
- You have 3 nickels, 2 dimes, and 8 quarters in your pocket. You want to know how much money you've got.
    - Write an algorithm to solve the problem.
    - From the algorithm, write a flowchart.
    - From the flowchart, write a computer program. Use sentences in your PRINT lines.

Lesson 16

THE REM STATEMENT

PERFORMANCE OBJECTIVES: Students will be able to:

- identify the function of the REM statement.
- explain the importance of the REM statement
- use the REM statement to document a program.

VOCABULARY: NEW: REM, documentation

REVIEW: PRINT, quotation marks

MOTIVATION:

- Let's take a look at this program:

```
10 LET A=1.50
20 LET B=2.25
30 LET C=0.75
40 LET D=5.00
50 LET E=A+B+C
60 LET F=.08 * E
70 LET G=E+F
80 LET H=D-G
90 PRINT H
```

- What does this program do? (It adds three numbers, multiplies the total by .08, and then adds this figure to the total. Then it subtracts the total from 5.00. Finally it prints out the result.)
- For what could this program be used? (It could be used to see how much change is left from $5.00 after purchases with sales tax have been made.)
- How could we have made the program easier to figure out? (We could have added PRINT statements.)
- Let's add PRINT statements to this program.
- Sometimes we don't want or need all of those PRINT lines in our output. What else could we do? (We could add programmers' comments or remarks that don't appear in the output.)

- These are called REM statements because they are programmers' REMarks.

AIM: How do we use the REM statment to document our program?

DEVELOPMENT:

1. Let's go back to our program with all of the PRINT lines:
   - How do we change these to REM statements? (We retype the lines to take out PRINT and the quotation marks, and we add REM as the first word.)
   - Change the first PRINT line.
   - Run the program and compare the output to the earlier program.
   - Change any other PRINT lines that are unnecessary to the user but essential to the programmer.
2. Why are REM statements, or program documentation, necessary to the programmer?
   - To understand what is happening step by step.
   - To alert another programmer to the purpose of each section.
   - To see the relationship of each part of the program to the others.
   - To make modification of program lines easier.
   - To understand the logic and sequence of the program.
3. In professional programs, the programmer begins by stating the title and function of the program and his own name or company.
   - How would you do this?
   - What happens when you need several lines in a row? (Each line must be numbered and start with REM.)
   - What effect do these lines have on the way the program runs? (None.)
   - Add a title, function, and your name to the program we've been using.

4. What do these REM statements do?

   -
     ```
     10 REM PROGRAM #7.  WRITTEN BY
     20 REM JANE DOE ON MAY 3, 1984.
     ```

   -
     ```
     70 REM X IS THE NUMBER OF
     80 REM RAINY DAYS THIS MONTH
     ```

   -
     ```
     50 REM C IS THE SUM OF A AND B
     ```

   -
     ```
     30 REM THE ANSWER WILL BE SHOWN
     40 REM BY THE FOLLOWING METHOD
     ```

5. Write REM statements for this program:

   ```
   10 REM
   20 REM
   30 PRINT "GOOD MORNING"
   40 REM
   50 PRINT "HOW ARE YOU?"
   60 END
   ```

6. Write a program with documentation for a list with your name, classes, and teachers.

APPLICATION:

- Write a program that displays the output below. Use REM statements, PRINT statements, and blank lines.

  ```
  REPORT CARD

  your name
  your address
  your city, state, and zip code

  SUBJECT       MARK

  ENGLISH        80
  MATH           75
  HISTORY        75
  LANGUAGE       70
  SCIENCE        80

  THE AVERAGE IS 76
  ```

SUMMARY:

- What is the purpose of documentation?

- How do we document programs in BASIC?

- What are some rules about the REM statement?

- How does the REM statment affect the execution, or running, of the program?

EXERCISES (ASSIGNMENT):

- Make corrections in these programs:

    -
      ```
      10 MY 12TH PROGRAM
      20 LET A=2
      30 LET B=3
      40 C=A+B
      50 PRINT C
      ```

    -
      ```
      THIS PROGRAM MULTIPLIES ONE NUMBER
      BY ANOTHER.  I WROTE IT FOR HOMEWORK.
      20 X=5
      30 Y=2
      40 Z=X*Y
      50 THE SOLUTION TO X TIMES Y
      60 PRINT Z
      ```

- Answer these questions:

    - What is the importance of the REM statment?

    - Give two situations in which you would use the REM statement.

- Write a telephone book program: List five friends, their complete addresses, and their phone numbers. Use REM statements to document your program.

Lesson 17

INPUT STATEMENT WITH NUMERIC AND/OR STRING VARIABLES

PERFORMANCE OBJECTIVES: Students will be able to:

- identify and use numeric and string variables.
- identify and use assignment statements.
- explain and use INPUT statements.

VOCABULARY: NEW: INPUT

REVIEW: numeric variable, string variable,
assignment statement, PRINT statement

MOTIVATION:

- Write a brief program which will multiply two numbers and print the result as a complete sentence.

  Acceptable program:

```
10  LET A = 4
20  LET B = 36
30  PRINT "The Product of"; A; "  Times"; B; "  is"; A*B
```

AIM: How do we use the INPUT statement to make our programs more flexible?

DEVELOPMENT:

1. In the program that we just wrote, what do we call "A" and "B?" (Variables.)
2. What is the function of the variables in a program? (They "hold" values and allow calculations to be performed.)
3. How does the computer know what "A" or "B" is equal to? (We "tell" the computer via assignment statement, i.e. LET A = 4.)

4. "A" and "B" had numeric values. How would we tell the computer that we want to use words instead of numbers with a variable? (We must use String Variables, i.e.: A$ as in LET A$ = "George")

5. Write a program that would make the computer print your name in a sentence.

   Acceptable Answer:

```
10  LET N$ = "Charles"
20  PRINT "My name is"; N$
```

6. Suppose your friend wanted the computer to print his or her name. Which line of this program would your friend have to change? (Line 10, assignment statement, would have to be changed so that N$ = new name.)

7. Suppose you wanted to change the values of "A" and "B" in our first program. Which lines would have to be changed? (Lines 10 and 20. New values would have to be assigned to "A" and "B.")

8. We are going to learn a new statement which will allow us to change the the value of numeric and string variables without rewriting program. Here is a program using this new statement:

```
10  PRINT "Type your age next to question mark."
20  INPUT A
30  LET Z = A + 10
40  PRINT "In ten years you will be  "; Z; "  years
  old."
```

- How did the computer know the value of "A"? (The user typed in value as requested in line 10.)

- What would be the value of "A" if each of the following were using this program: your kid brother/sister, your mother/father, your grandparent? ("A" would have the value typed in by each of these users.)

9. Why would this program save us the trouble of rewriting each time a different person used the program? (This program is "flexible" because the value of "A" can be entered by user; it can be changed each time program is RUN.)

10. Which line made this program more flexible? (Line 20, the INPUT statement.)

11. Let's write a definition of this very important INPUT statement:

- The INPUT statement allows us to assign a value to a variable as we use the program.

- The INPUT statement places a question mark on the screen and makes the program wait until the user has entered a value. The program then continues.

- The value assigned to the variable can be different each time the program is run.

12. What would appear on the screen as the output of this program?

```
10  PRINT "Type your first name next to the question
          mark."
20  INPUT N$
30  PRINT "Pleased to meet you,  "; N$
```

(Pleased to meet you, student's name)

- Note: unless the Clear/Screen function is used, line 10 will also appear. See explanation in APPLICATION section.

13. What type of variable did we use in this program? (String variable.)

14. We see that INPUT statements can be used with either string or numeric variables. We have one more thing to learn. Compare this program to the program we just did:

```
10  INPUT "Type your first name next to the question
          mark"; N$
20  PRINT "Pleased to meet you,  "; N$
```

15. What differences are there between the two programs? (This program uses fewer lines by combining the old lines 10 and 20. Be sure students understand the syntax of new line 10.)

APPLICATION:

1. Write a program which will ask a person's name, address, and phone number and which will then print out this information in sentence form.

Acceptable Answer:

```
10  INPUT "What is your name"; N$
20  INPUT "Where do you live"; A$
30  INPUT "What is your phone number"; T$
40  (clear-screen command -- see * below)
50  PRINT N$; " lives at  "; A$
60  PRINT N$; "'s telephone number is  "; T$
70  END
```

* To clear screen:

- Apple = HOME
- TRS-80 = CLS
- PET = PRINT "Shift, CLEAR/HOME"

2. Write a program which will compute the cost of any item when given the item, price, and number purchased.

   Acceptable Answer:

```
10  INPUT "What did you buy"; I$
20  INPUT "What does each cost"; C
30  INPUT "How many did you buy"; N
40  (clear-screen, see * in previous problem)
50  LET T = C*N
60  PRINT N; I$; "s cost"; T
```

3. Write a program which asks for a person's name and birthday, causes the screen to clear, and then prints an output in sentence form.

SUMMARY:

- What are the two methods by which the computer can be assigned values for variables used in a program?
- What are the similarities and differences between Assignment Statements and INPUT statement?
- What are the two variations of INPUT statements?
- When do you use each? Why?

EXERCISES (ASSIGNMENTS):

- Write a program which will collect information about a person's favorite record, movie, television show, and performer(s). Your output should be in sentence form and include the name of the person using the program. The screen should clear after the questions are asked.

- Write a program which will calculate the amount of change a person would receive from a dollar after purchasing an item. Output should include person's name, cost of item, and change given, all in sentence form. (Challenge: Change this program to calculate the change received when items cost more than one dollar and person uses high denomination bills (i.e. five-dollar bill, ten-dollar bill, etc.)

- Write a program which will add the prices of several items. Output should include names of items, number of items purchased, cost of each item, and total amount spent, all in sentence form.

Lesson 18

INPUT STATEMENTS USING PROMPTS

PERFORMANCE OBJECTIVES: Students will be able to:

- explain the need for user-prompts in an interactive program.
- create clear and adequate prompts in a student-written program.

VOCABULARY: NEW: Prompt, Delimiter, (Interactive)

REVIEW: INPUT, String Variable, Numeric Variable, Value

MOTIVATION:

- We are going to perform an experiment. The first step is to type this program at your keyboard:

```
10  PRINT "Type Three Numbers"
20  INPUT A,B,C
30  LET Z = A + B + C
40  PRINT "Total is "; Z
```

- Do each of the following steps. For each step, observe the screen and note your observations on paper:
- Type RUN. Type, exactly as shown here, the numbers 1 2 3 and press ENTER (RETURN). What is the screen output?
- Type RUN. Type, exactly as shown 1 2 3 <u>without</u> pressing ENTER (RETURN). What is the screen output?
  - Type RUN. Type 1,2,3 Record output.

Ask class to report findings. These should be as follows:

- "Type Three Numbers" appears.
- "??" appears; the computer is waiting for next number(s).

- Nothing happens as ENTER (RETURN) was not pressed.

- Correct answer, 6 appears in output line with the words, "Total is".

- Is the computer "stupid?" (It is not "stupid" but a machine expecting a particular response in a particular form and reacting if the response is not in this correct form.)

- This program required the user to INPUT data in a certain way. What do you think the point of this lesson will be? (Elicit AIM)

AIM: How can we give proper instructions to the user so that the computer can understand our INPUT?

DEVELOPMENT:

1. To review concept of variables, assignment of values, and INPUT:

    - In the program above, what is the correct name for "A," "B," etc? (Numeric variables.)

    - What other kind of variable have we studied? (String variables)

    - How are numeric and string variables different?

    - What is the purpose of using a variable? (Variables hold an assigned value for the purpose of calculations.)

    - What is the relation between INPUT and the value of a variable? (The response to an INPUT assigns the value to a variable. The user enters the value.)

2. Explain: When a program uses INPUT, it becomes an "interactive" program because it allows the user to "interact" with the computer by entering data rather than using pre-assigned values.

3. How did line 20 inform the computer to expect exactly 3 separate values? ("INPUT A,B,C" tells the computer to expect 3 separate and distinct numbers.)

4. How does the computer tell the user to type in a value? (It places a "?" on the screen and waits.)

   - If the computer expects 3 separate values and the user types only 1 and presses ENTER (RETURN), what happens? (The computer places a "?" or "??" on the screen to show it is waiting for next value.)

   - How many times will it place "?" or "??" on the screen? (Once each for as many variables as specified in INPUT statement.)

5. When you did the third step in our experiment (see motivation), you typed: 1,2,3. There was no "??" and you got the correct answer. Why? (Typing 1,2,3 with commas as shown satisfied the computer's expectation of three separate values and the program proceeded to execute.)

   - Explain: The commas are called "Delimiters."

   - Delimiters separate numbers so that the computer "sees" them as separate and distinct values.

6. How could we teach the user to use Delimiters so that the program runs faster and so that the correct number of values is entered? (Write a message or hint in the program.)

   - What might the message have said in our experiment program? ("Type 3 numbers, separated by commas.")

   - Where would this go in the program? (Change Line 10 to read: 10 PRINT "Type 3 numbers, separated by commas.")

7. Explain: This sort of message is called a Prompt. It prompts the user to respond so that the computer can understand the INPUT. It makes the program run properly and avoids confusion.

8. Our experiment program needed another prompt. In the second step you entered some numbers but you were not told to press ENTER (RETURN). What happened? (Nothing, because values had not been "ENTERED" into storage.)

   - How could we have made sure that the user would press ENTER (RETURN)? (By writing a prompt into the program.)

- What might this prompt have been? (15 PRINT "Press ENTER (RETURN) when finished.")

- Why would this have been helpful to a novice user? (It would have provided direction, avoided confusion, frustration, etc.)

9. We have learned that the computer can accept words as values. What do we call this kind of variable? (String variable, such as A$ = "John")

   - What might happen if the computer expected a string variable and the user typed a number, or vice-versa? (The user would get an error message, such as "Re-Do.")

   - How could we help the user to avoid getting such an error message? (Write in a prompt.)

   - What might such a prompt say? (PRINT "type full words only" etc.)

10. How are these prompts different from REM statements?

APPLICATION:

- Write a program which will calculate the age differences among three or more people. Be sure to add prompts which will cause the user to enter words where they are required and numbers where they are required. The prompts should ask for the persons to be entered in order of decreasing age.

  Acceptable Answer:

```
 10 PRINT "List names of three people"
 20 PRINT "With oldest first, youngest last."
 30 PRINT "Type 3 names, separated by commas. Press
 ENTER(RETURN)."
 40 INPUT A$, B$, C$
 50 PRINT "Now type the age of each person in years."
 60 PRINT "Type 3 numbers, separated by commas.  Press
 ENTER (RETURN)."
 70 INPUT A,B,C
 80 LET X = A-B
 90 LET Y = B-C
100 LET Z = A-C
110 PRINT A$; " is "; X; "years older than "; B$
120 PRINT B$; "is "; Y; "years older than "; C$
130 PRINT "But "; A$; " is "; Z; "years older than "; C$
```

SUMMARY:

- What do we call the messages which tell the user what to type in an interactive program?
- How do these prompts help the user?
- How do they speed up the program?

EXERCISES (ASSIGNMENTS):

- Write a program which will require the user to INPUT the names and telephone numbers of five people. Make it PRINT a list of each person's name accompanied by the correct phone number. Use proper spacing between lines, REM statments, etc. Be sure to add prompts for two types of variables.
- Write a program which will ask the user to INPUT the main dish for a full week's worth of ideal school lunch menus. The output should be in the form:

  ```
  Day #____      Name of Day    Main Dish     Price
  ```

  Use headings and proper line spacing. Be sure to use prompts throughout.
- Write a program which would create a Christmas shopping list. The user would INPUT the items desired, name of recipient, and estimated cost. The output would be in this format:

  ```
  Person    Gift Item     Estimated Cost
  ```

  Use headings and spacing. This program should also provide a final total line. Use prompts, of course!

Lesson 19

READ/DATA STATEMENTS (ONE VARIABLE)

PERFORMANCE OBJECTIVES: Students will be able to:

- properly provide data using a READ/DATA Statement.
- distinguish between the "INPUT" and "READ/DATA" statements.
- write a simple program using READ/DATA.
- identify common errors made in using READ/DATA statements.

VOCABULARY: NEW: READ/DATA
REVIEW: INPUT

MOTIVATION:

- Write a program that, when given the price of an item, will calculate the sales tax and will then print out the given price, the calculated tax, and the total cost (price + tax) on separate lines.

  (acceptable answer)

```
10 REM PRICE + TAX = TOTAL
20 REM P = PRICE, T = TAX, C = TOTAL COST
30 INPUT P
40 LET T = P * .0825
50 LET C = P+T
60 PRINT "PRICE OF ITEM = "; P
70 PRINT "NYC TAX = "; T
80 PRINT "TOTAL COST = "; C
90 GOTO 30
```

- What is the purpose of line 30? (The computer waits for value to be entered.)
- What is the purpose of line 90? (It loops back for a new value to be entered.)
- Explain: The values entered for P in line 30 are also known as data or information needed by the program.

- Suppose a user had five separate items. How many times would he have to enter price data? (Five times, once for each item.)

- How could we make this program operate faster? (Find a way to enter all prices at once.)

- That is the purpose of this lesson. (Refer to AIM below.)

AIM: How can we provide a program with more than one item of data at a time?

DEVELOPMENT:

1. Here is a new version of the program above. Which lines have been changed?

```
10 REM PRICE + TAX = TOTAL
20 REM P = PRICE, T = TAX, C = TOTAL COST
30 READ P
40 LET T = P * .0825
50 LET C = P + T
60 PRINT "PRICE OF ITEM =  "; P
70 PRINT "NYC TAX =  ";  T
80 PRINT "TOTAL COST =  ";  C
90 DATA 4.00, 3.75, 8.50, 10.00, 6.25
100 GOTO 30
```

(30 INPUT P now reads 30 READ P. 90 DATA 4.00, 3.75, 8.50, 10.00, 6.25 has been added, causing GOTO to drop one line lower)

- How many Total Costs will be printed out? (Five)

- From where is the program getting the five prices necessary to create five outputs? (Line 75 DATA 4.00, 3.75, etc.)

- Explain: Lines 30 and 75 are the secret to making this a faster program. Together they make up a new programming statement called READ/DATA. Both the READ and the DATA lines must be present.

2. In either version, what does the variable P represent? (Price of item; i.e. price data.)

   - In either version, what does the variable P represent? (price of item; i.e., price data)

- How does new version obtain price data? (It reads from the DATA line.)

- Explain: Statement READ P tells the computer to look for a DATA line. It picks out the first value in the line. Program performs calculations and produces output. Statement GOTO 30 loops program back to READ P. There is an internal "pointer" which forces computer to use the next value in line as value for P. Process continues until all values in DATA line are used.

3. How many times will computer go through this processing loop? (Until all data items have been assigned to variable P.)

   - This particular program will perform five separate processing loops. What will happen on the sixth try? (No more data is available.)

   - Explain: If all the data has been used, the computer will place an "Out of Data" message on the screen. In a future lesson we will learn a method to avoid this unnecessary data message (using IF/THEN).

4. Suppose we wanted the computer to produce a brief thank-you note to each of five friends named Ralph, Jenny, Juan, Migdalia, and Wanda. Write a brief program to do this, using the READ/DATA statement.

   (acceptable program)

```
10 READ N$
20 PRINT "DEAR  ";  N$
30 PRINT "           THANK YOU FOR THE PRESENT."
40 PRINT
50 PRINT "                      LOVE,"
60 PRINT "                        MARY"
70 DATA RALPH, JENNY, JUAN, MIGDALIA, WANDA
80 GOTO 10
90 END
```

5. What are the differences in the READ/DATA between this program and the sales tax program? (READ N$; DATA line had string instead of numeric values)

   - What are the similarities between the two programs? (In both, values in the DATA line are separated by commas.)

6. What rules have we derived concerning this new programming statement?

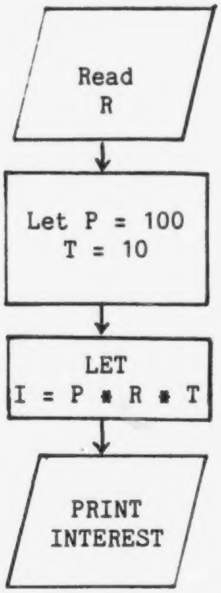
- READ is followed by numeric or string variable.
- DATA must appear in the program, usually close to end.
- DATA values must be separated by commas, except after the last.
- READ can never be used without DATA; vice-versa.

APPLICATION:

Using READ/DATA, write a program using the formula
Interest = Principal x Rate x Time
With Principal = $1,000
FIXED
Time = 10 Years
and the following Rates: .08, .09, .10, .11, .12

Print out the different interest earned at each rate.

Flowchart:



Data: .08, .09, .10, .11, .12

Program:

```
10  READ R
20  LET P = 1000
30  LET T = 10
40  LET I = P * R * T
50  PRINT "INTEREST = "; I
60  DATA .08, .09, .10, .11, .12
70  GOTO 10
```

SUMMARY:

- What are the differences between the INPUT statement and the READ/DATA statement?

- What are the advantages of READ/DATA? INPUT? When would you use each?

EXERCISES (ASSIGNMENT):

- Find and correct the error in each of the READ/DATA statements below:

  -
    ```
    10  READ A
    20  DATA "JOE", 20, "TOM", 15
    ```

  -
    ```
    10  DATA  6  4  10
    20  READ P
    ```

  -
    ```
    10  READ 5
    20  DATA 12, 31, 41
    ```

  -
    ```
    READ 5
    DATA  4, 0, 1
    ```

b. Rewrite each of these programs using READ/DATA. Make up pieces of data.

```
10  LET A = 3
20  INPUT B
30  LET C = A * B
40  PRINT C
50  GOTO 20
```

```
10  INPUT A$
20  PRINT "YOUR NAME IS" A$
30  GOTO 10
```

C. What is the output of the following program?

```
10  READ A
20  PRINT A
30  READ V
40  PRINT V
50  LET C = A * V
60  PRINT C
70  DATA 4, 5
80  END
```

d. Write a program that will read five test marks and print the average. Use READ/DATA.

Lesson 20

READ/DATA STATEMENTS (MULTIPLE VARIABLES)

PERFORMANCE OBJECTIVES: Students will be able to:

- provide values for more than one variable using a READ/DATA statement.
- identify errors commonly found when using READ/DATA with more than one variable.
- write a program with more than one variable using READ/DATA statements.

VOCABULARY: REVIEW: READ/DATA, value, variable, pointer

MOTIVATION:

- Examine Program (A) and Program (B) for similarities and for differences:

Program (A)

```
10 REM THIS PROGRAM CALCULATES INTEREST
20 REM P = PRINCIPAL, R = RATE, T = TIME
30 REM I = INTEREST (P*R*T)
40 INPUT P
50 LET R = .08
60 LET T = 10
70 LET I = (P*R*T)
80 PRINT "INTEREST =   "; I
90 GOTO 40
```

Program (B)

```
10 REM THIS PROGRAM CALCULATES INTEREST
20 REM P = PRINCIPAL, R = RATE, T = TIME
30 REM I = INTEREST (P*R*T)
40 READ P
50 LET R = .08
60 LET T = 10
70 LET I = (P*R*T)
80 PRINT "INTEREST =   "; I
90 DATA 1000, 550, 725, 364, 930
100 GOTO 40
```

- What is the purpose of READ/DATA in Program (B)? (It avoids the need for user input of data.)

- Why is this a possible advantage? (It is faster, more efficient, etc.)

- Suppose instead of a fixed rate of 08 we could earn 09, 10, or even a 12 percent rate. How could we cause Program (A) to consider not only a variety of principal but also of interest rates? (We would need a separate variable for rate; i.e., INPUT P,R.)

- Suppose we wished to run Program (A) with a varying number of years. How many variables would we need in our INPUT statement? (Three; i.e., INPUT P,R,T.)

- But we've discovered that the INPUT in Program (A) causes the program to wait for the user to enter data. We know that READ/DATA is more efficient. So far we've used READ/DATA with only one variable. What, therefore, would be the purpose of this lesson? (Refer to AIM below.)

AIM: How do we use READ/DATA with multiple variables?

DEVELOPMENT:

1. Suppose we wanted to run Program (B) for a range of different Principals, different Rates, and different Times. How many variables would we be working with? (Three: P, R, and T.)

    - How would three variables in the READ statement appear? (READ P,R,T.)

    - READ is half of a pair. What is the other half? (DATA.)

    - How should we arrange the DATA line so that the computer can find the correct values for P, R, and T? (The values in the DATA line must be in groups of three and must be in proper order corresponding to the variables.)

    - How does the computer's "internal pointer" work with a READ/DATA statment? (The pointer advances on each "pass" thus guaranteeing that the correct value is assigned to the variable each time around.)

    - Explain: with multiple variables, the pointer feeds sets of values to variables. The number of values fed on each pass is determined by the number of variables in the READ line.

2. Suppose a program had the following two lines, among others:

   ```
   READ X,Y,Z
   DATA 20, 2, 12, 30, 3, 13, 40, 4, 14, 50, 5, 15
   ```

   - How will the computer sort this out? (It knows to expect three values on each pass. On the first pass, the first three values are assigned to the variables. On the second pass, the next set of three values is assigned, etc.)

3. What might happen if the above DATA line looked like this:

   ```
   DATA 20, 30, 40, 50, 2, 3, 4, 5, 12, 13, 14, 15
   ```

   (It would still access sets of three values, but the wrong values would be assigned to X, Y, and Z. Output would be wrong or nonexistent.)

4. What might happen if your DATA line looked like this:

   ```
   DATA 20, 2, 12, 30, 3
   ```

   (The program would cease operating because there are not enough values on second pass.)

5. Considering the above, what are some important rules to remember when using READ/DATA with multiple variables?

   - The order of values in DATA must match the order of variables in READ.

   - The total number of values in DATA must be exactly divisible by the number of variables in READ: i.e., 3 variables require either 3 or 6 or 9 or 12 etc. values in DATA line.

   - The values in DATA must be separated by commas, but no comma is used after the last item.

   - Explain that, for reason of screen width, it might be necessary to use more than one DATA line. Each DATA line must follow rules above.

6. Rewrite Program (B) using READ/DATA with multiple variables: Principal, Rate, and Time. Make up three sets of three values apiece for the DATA line.

Acceptable program:

* lines 10, 20, 30 are the same as in earlier program(s)

```
 40  READ P,R,T
 50  LET I = (P*R*T)
 60  PRINT P; "DOLLARS AT   "; R; " RATE FOR   "; T;
     " YEARS"
 70  PRINT "EQUALS INTEREST OF   "; I
 80  DATA 1000, .08, 5, 5000, .05, 8, 3500, .12, 9
 90  GOTO 40
100  END
```

7. What other type of variable is there besides numeric? (String variable.)

   - What would the READ line show if your first of three variables were a string varible? (READ N$,X,Y)

   - What would the DATA line show to match this READ line? (DATA TOM, 20, .03, DICK, 80, .14, HARRY, 35, .10)

APPLICATION:

Write a program to list 5 teams, the number of games each has won and lost, and the winning percentage of each.

	W	L
Giants	20	10
Cubs	10	31
Mets	40	12
Phils	22	23
Cards	41	21

Program:

```
10 READ T$, W, L
20 LET PCT = W/(W+L)
30 PRINT T$, W, L, PCT
40 DATA GIANTS, 20, 10, CUBS, 10, 31
41 DATA METS, 40, 12, PHILS, 22, 23
42 DATA CARDS, 41, 21
50 GOTO 10
```

NOTE:

- More than one set of data can appear on a data line.
- Commas separate data except at the end of the data line.
- The order of the data here is: string, number, number.

SUMMARY:

- If a program uses a READ line, what other line must appear in the program?
- Why must a programmer be careful when arranging the values in a DATA line?
- Explain how the internal pointer handles sets of DATA values for multiple variables.

EXERCISES (ASSIGNMENT):

- Find and correct any errors in each READ/DATA statement below:

-
  ```
  10 READ X, Y, W, Z
  30 DATA 5, 10, 15
  ```

-
  ```
  10 READ X; Y; Z
  20 DATA 2, 4, 6
  ```

-
  ```
  10 DATA $2, $3, $4
  20 READ D, E, F
  ```

-
  ```
  10 READ M$, N
  20 DATA 32, "JOHN", 41, "SAM"
  ```

-
  ```
  10 READ A, B
  20 DATA 2, 3, 4, 5, 6
  ```

- Rewrite this program using READ/DATA - Make up four pieces of data.

```
10  INPUT A, B, C
20  LET R = (A+B+C)/3
30  PRINT R
40  GOTO 10
```

- Give the output for these programs:

-
  ```
  10 DATA 51, 17, 4, 3
  20 READ Q, R
  30 PRINT "AVE OF Q AND R IS "; (Q+R)/2
  40 GOTO 20
  ```

-
  ```
  10 READ A,B,C
  20 DATA 2,5
  30 DATA 3,1,7,8
  40 PRINT A+B+C
  50 GOTO 10
  ```

- Write a program to print the area and perimeter of a rectangle whose lengths and widths are:

L	W
62 cm	38 cm
71 cm	92 cm
42 cm	100 cm

Lesson 21

FLOWCHARTS AND PROBLEM SOLVING WITH CONDITIONALS

PERFORMANCE OBJECTIVES: Students will be able to:

- identify, write and apply symbols of inequality in standard form.
- identify the flowchart symbols for decision and connector.
- interpret and trace a flowchart involving conditional statements.
- create flowcharts to solve problems involving conditionals.

VOCABULARY: NEW: relational operators (inequality symbols), conditional branching, decision box, connector

REVIEW: $<, >, \leq, \geq, =, \neq$

MOTIVATION:

- To be an "Honor Roll Candidate" a student needs an average of 90 or above in five major subjects.
- If your grades are: 100, 80, 91, 95, and 88, are you an "Honor Roll Candidate"?
- How do we figure out this problem?
- Why might we want to create a computer program to solve the honor roll problem?
- Let's try to write the steps in words.
  - Input the 5 grades.
  - Calculate the average.
  - If the average is 90 or more, print "Honor Roll Candidate."
  - If the average is less than 90 do not print "Honor Roll Candidate."

- What do we call the symbols we use for less than, greater than, etc.? (relational operators)
- Let's create a chart to show these symbols:

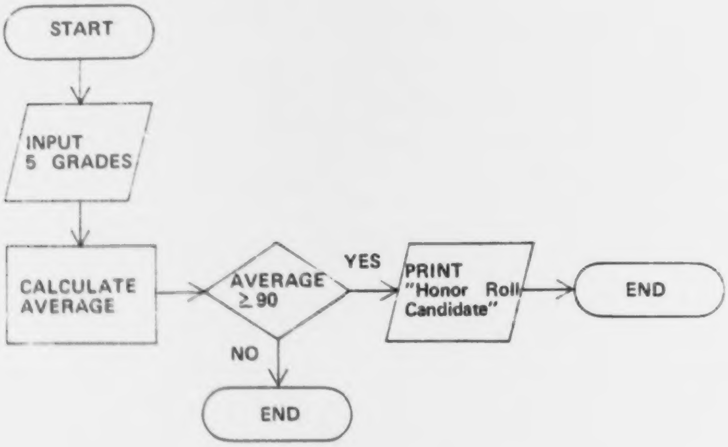
Verbal Meaning	Arithmetic Symbol	BASIC Symbol
Less than	$<$	<
Greater than	$>$	>
Equal to	$=$	=
Less than or Equal to	$\leq$	<=
Greater than or Equal to	$\geq$	>=
Not equal to	$\neq$	<>

- What is the next step before we write a computer program for the honor roll problem? (We must write a flowchart.)

AIM: How do we prepare flowcharts for problems involving conditional statements?

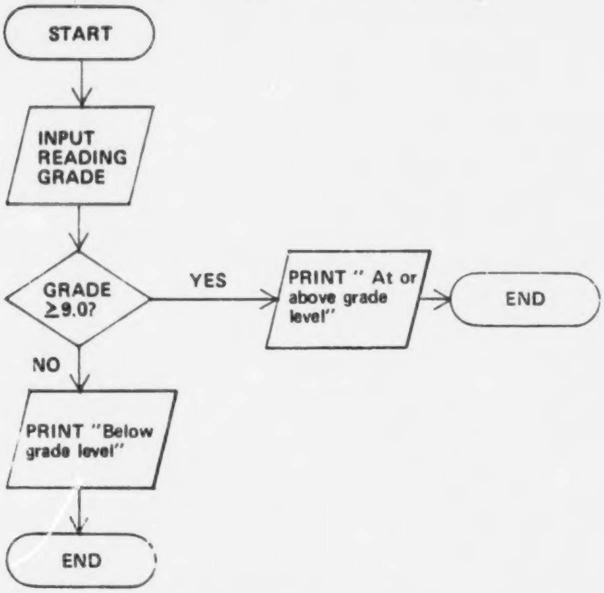
DEVELOPMENT:

1. Develop flowchart.


START
INPUT 5 GRADES
CALCULATE AVERAGE
AVERAGE ≥ 90
YES
PRINT "Honor Roll Candidate"
END
NO
END

2. Notice the use of the new "diamond" symbol. This is called a decision box. If the condition in the box is true, follow the "YES" arrow; if false, the "NO" arrow.

3. Create a flowchart to solve the following problem? Students want to know if they are reading above or below grade level. Grade level is 9.0.


START
INPUT READING GRADE
GRADE ≥9.0?
YES
PRINT " At or above grade level"
END
NO
PRINT "Below grade level"
END

4. Write the following statements using arithmetic and BASIC symbols:

   - 100 equals 10 times 10.
   - 50 less than 100.
   - 50 is greater than 25.
   - an unknown number does not equal 100.
   - an unknown number is less than or equal to 50.
   - an unknown number is greater than or equal to 25.

   - Write the following statements using arithmetic and BASIC symbols:

     - 5 is greater than 4.
     - 3 is less than 5.
     - 10 equals 5 times 2.
     - 10 is not equal to 5.
     - an unknown number is less than or equal to 10.
     - an unknown number is greater than or equal to 10.

- Create a flowchart to solve the following problem:

  You have $5.00 to spend. You have selected 3 items. You must add sales tax to these items. You want to know if you have enough money to buy them all.

SUMMARY: How do the flowcharts we created today differ from previous flowcharts?

EXERCISES (ASSIGNMENT):

- State the following in words:
  - $16 \neq 7 x 2$
  - 18 <> 8*2
  - $10 = 5 + 5$
  - $X < 8$
  - $10 > 8$
  - x < = 10
  - 5 > = x
- Write a flowchart to PRINT the larger number of two unequal numbers.
- Write a flowchart for a program to calculate the salary earned if the rate of pay is $3.00 for the first 40 hours and $4.50 for any additional hours.
- Write a flowchart to find the cost of a cab ride if the first mile costs $5.00 and each additional 1/2 mile costs $.50.

Lesson 22

PROGRAMS WITH CONDITIONAL BRANCHING (IF/THEN STATEMENTS)

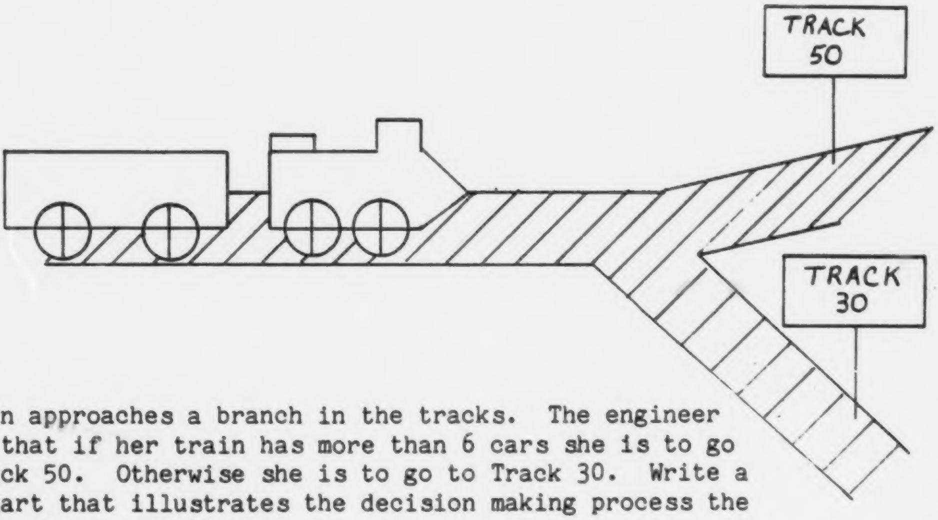
PERFORMANCE OBJECTIVES: Students will be able to:

- determine whether a given equality or inequality is satisfied.
- distinguish between valid and invalid uses of the IF/THEN statement.
- identify, write and apply the IF/THEN statement.
- write simple programs involving IF/THEN.

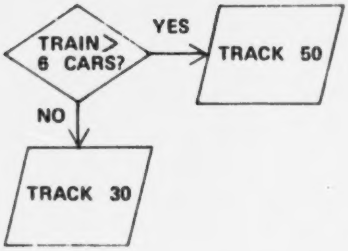
VOCABULARY: NEW: IF/THEN statement

REVIEW: conditional branching, relational operators

MOTIVATION:



- A train approaches a branch in the tracks. The engineer knows that if her train has more than 6 cars she is to go to Track 50. Otherwise she is to go to Track 30. Write a flowchart that illustrates the decision making process the engineer has to go through.
- Discuss the solution.
- Review relational operators in both standard and BASIC notation.



- Review the diamond shaped decision box of a flowchart. Remind the students that all questions in a decision box must be of the yes-no variety and have two branches coming out.

- What is the next step? (Elicit AIM.)

AIM: How do we write programs involving conditional branching?

DEVELOPMENT:

1. Discuss how the flow chart from the MOTIVATION becomes a BASIC program.

```
10 READ NAME$, CARS
20 IF CARS > 6 THEN 50
30 PRINT NAME$ " ARRIVING ON TRACK 30"
40 GOTO 10
50 PRINT NAME$ " ARRIVING ON TRACK 50"
60 GOTO 10
70 DATA F, 8, GG, 5, M, 6
```

2. Explain the general form of the IF/THEN statement: IF (condition) THEN (line #).

3. Explain the following examples:

```
· 40 IF K = J THEN 50
  50 IF A + B > 6 THEN 20
  60 IF R > = 10 THEN 100
  70 IF A*B > A + B THEN 90
```

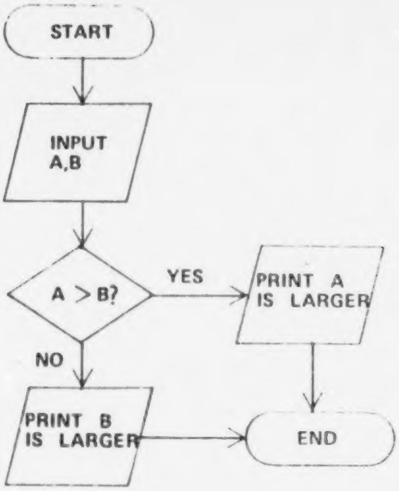
4. How does the computer check the condition? (If the response to the condition is yes, the computer goes to the line number indicated in the THEN part of the statement. If the response to the condition is no, the program continues with the next line number.)

   - Carefully walk through the logic of the program using the data given in line 70 of the program.

APPLICATIONS:

- Prepare a flowchart and write the program to output the larger of two unequal numbers.

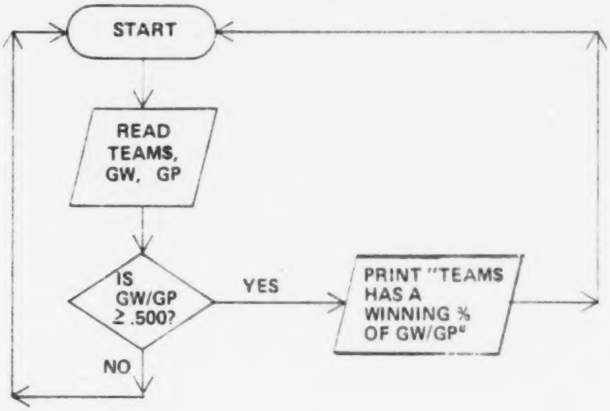
- Answers:


START
INPUT A,B
A > B?
YES
NO
PRINT A IS LARGER
PRINT B IS LARGER
END

```
10 INPUT A, B
20 IF A > B THEN 50
30 PRINT B; "IS LARGER"
40 GOTO 60
50 PRINT A; "IS LARGER"
60 END
```

- Write a flowchart and the program that will output the name of those teams with a winning percentage of .500 or more. You are given the number of games played.

- Answers:


START
READ TEAMS, GW, GP
IS GW/GP ≥ .500?
YES
NO
PRINT "TEAMS HAS A WINNING % OF GW/GP"

```
• 10 READ TEAM$, GW, GP
  20 IF GW/GP = .500 THEN 40
  30 GOTO 10
  40 PRINT TEAM$ "HAS A WINNING
     % OF1 1"1GW/GP
  50 GOTO 10
  60 DATA
```

- Note: When we write the IF/THEN statement in the program, the line number after the THEN may have to be written later, when the program is more fully developed.

SUMMARY:

How is the IF/THEN statement executed when the condition is (1) true, (2) false?

EXERCISES (ASSIGNMENT):

- Answer <u>yes</u> or <u>no</u> to indicate whether each of the following is a valid IF/THEN Statement:
  - `20 IF A > B THEN GOTO 70`
  - `60 IF M - N > = 0 THEN 100`
  - `80 IF X*Y<>X + Y THEN 20`
  - `100 IF D + 7 THEN 40`
  - `70 IF X < 3 THEN 70`
- Determine if the following inequalities are true or false. Use the values A = 7, B = 5, C = 10, D = 15.
  - `2 * A - B >C`
  - `A/B >= D/C`
- What is wrong with this program?
  - `10 INPUT A`
  - `20 IF A >10 THEN 30`
  - `30 PRINT A "IS LARGER THAN 10"`
  - `40 END`
- Write a program that will INPUT the name of a car and its cost. If the cost is more than or equal to $10,000, print out that the car is a luxury car. If the cost is less than $10,000, print out that the car is an economy car.
- Write a program that will print the names of all salespeople whose gross sales were more than $1,000 for the week. Use the following data:

Hopkins	$7,500
Baker	$ 650
Meyer	$9,000
Ryan	$ 400

Lesson 23

WRITING PROGRAMS WITH IF/THEN STATEMENTS AND NUMERIC VARIABLES

PERFORMANCE OBJECTIVES: Students will be able to:

- use a trailing value in a DATA statement to avoid an "out of data" error statement.
- write a program using IF/THEN statements that compare two numerical variables.
- write a program involving conditions with upper and lower bounds.

VOCABULARY: NEW: Flag
REVIEW: IF/THEN statement, compound inequality

MOTIVATION:

- What is the output for this program?

```
10 READ X
20 IF X = 999 THEN 60
30 PRINT X
40 GOTO 10
50 DATA 2, 4, 6, 5, 999
60 END
```

- What is the role of the 999?
  - Define this kind of trailing value in a DATA statement as a flag.
  - What would happen to the output if line 20 were removed? (OUT OF DATA error message appears.)
  - Why should we avoid OUT OF DATA messages?
- Consider the following problem: Acme Company pays $4.80 per hour to its part-timers and Ajax Company pays $5.20 per hour. Write a flowchart that will compare the salaries of Joe and Roy week by week based on this chart:

	ACME CO.	AJAX CO.
	JOE	ROY
Week 1	12 Hours	8 Hours
Week 2	14 Hours	10 Hours
Week 3	10 Hours	12 Hours
Week 4	9 Hours	19 Hours
Week 5	20 Hours	15 Hours

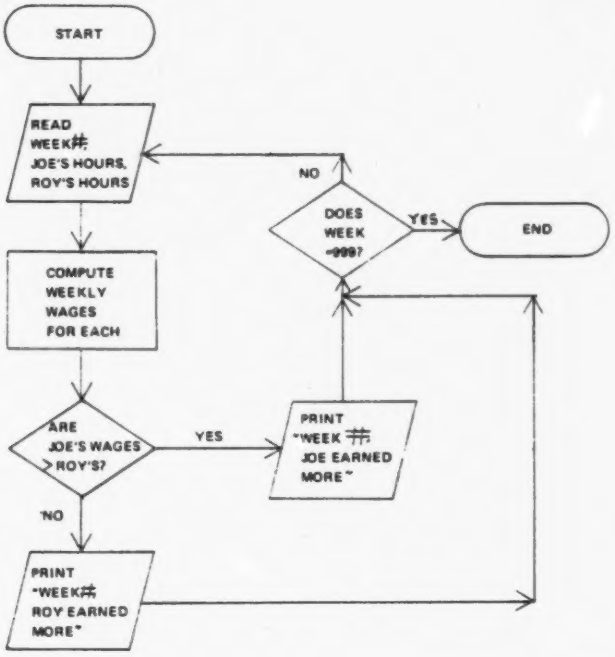
- What is our next step? (We have to construct a program from the flowchart.)

AIM: How do we write a program comparing numerical variables?

DEVELOPMENT:

1. Review the three parts to every program: INPUT, PROCESS, OUTPUT. What new process have you just learned? (Making comparisons, i.e., the IF/THEN statement.)

2. Use the flowchart to assist with the writing of this program:


START
READ WEEK#; JOE'S HOURS, ROY'S HOURS
COMPUTE WEEKLY WAGES FOR EACH
ARE JOE'S WAGES > ROY'S?
YES
NO
PRINT "WEEK #: JOE EARNED MORE"
PRINT "WEEK#: ROY EARNED MORE"
DOES WEEK =999?
YES
NO
END

3. How do we write the program?

```
  5 READ W, J, R
 10 IF W = 999 THEN 110
 20 JW = J * 4.80
 30 RW = R * 5.20
 40 IF JW  > RW THEN 70
 50 PRINT "WEEK"; W; "ROY EARNED MORE"
 60 GOTO 5
 70 PRINT "WEEK"; W; "JOE EARNED MORE"
 80 GOTO 5
 90 DATA 1, 12, 8, 2, 14, 10, 3, 10, 12
100 DATA 4, 9, 19, 5, 20, 15, 999
110 END
```

- Note: Line numbers associated with GOTO or THEN statements should be left blank until all line numbers are established.

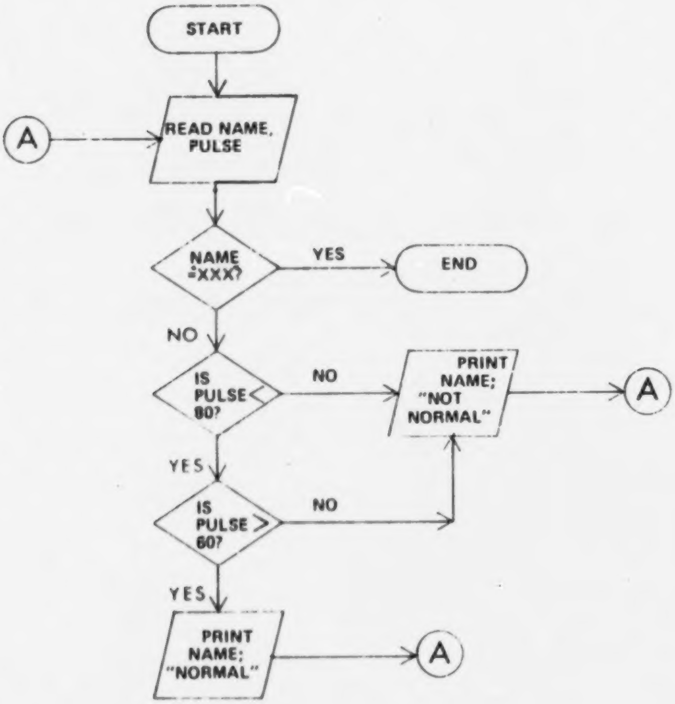
APPLICATION:

- Prepare a flowchart and write a program for the following problem:

  If a person's pulse is between 60 and 80 beats per minute, he is said to be normal. Write a program to check if these people are within normal range.

Name	Pulse
Mr. Adams	62
Mr. Bell	78
Ms. Cox	84
Ms. Dell	58

- Flowchart:


START
A
READ NAME, PULSE
NAME =XXX?
YES
END
NO
IS PULSE< 80?
NO
PRINT NAME; "NOT NORMAL"
A
YES
IS PULSE > 60?
NO
YES
PRINT NAME; "NORMAL"
A

- Program:

```
 10 READ NAME$, P
 15 IF NAME$ = "XXX" THEN 110
 20 IF P< 80 THEN 50
 30 PRINT NAME$; " NOT NORMAL"
 40 GOTO 10
 50 IF P> 60 THEN 80
 60 PRINT NAME$; " NOT NORMAL"
 70 GOTO 10
 80 PRINT NAME$; " NORMAL"
 90 GOTO 10
100 DATA ADAMS, 62, BELL, 78, COX, 84, DELL, 58, XXX, 999
110 END
```

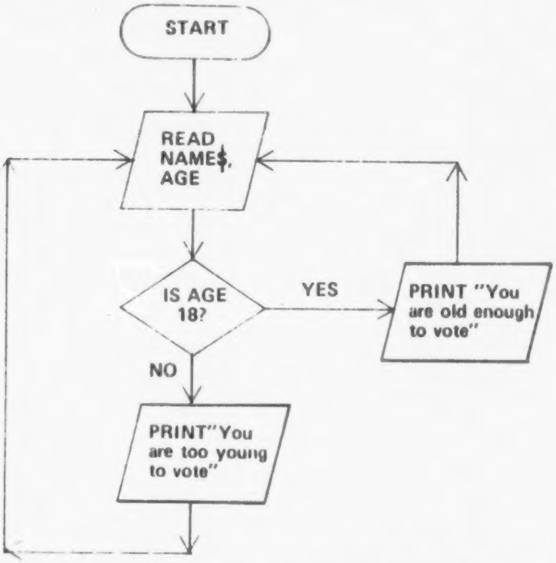
SUMMARY:

What is a flag? How is it used?

What are some key points to look for when writing a program with conditionals?

EXERCISES (ASSIGNMENT):

- Which are valid IF/THEN statements?
  - 10 IF A = B THEN 60
  - 30 IF X = Y = Z THEN 10
  - 40 IF X * 4 = Z THEN 200
- Add a flag to this flow chart:


START
READ NAME$, AGE
IS AGE 18?
YES
PRINT "You are old enough to vote"
NO
PRINT "You are too young to vote"

- Write a program from the new flowchart. Use DATA ROBERT, 20, EDDIE, 15, ROSA, 18.
- Write a flowchart and program. Any person who is between 40 and 60 is "middle aged." Write a program to determine which of these people are "middle aged."

Name	Age
Mr. Ross	35
Ms. Finn	40
Mr. Jones	45
Ms. Smith	70

Lesson 24

PROGRAM TRACE WITH UNCONDITIONAL BRANCHING (GOTO)

PERFORMANCE OBJECTIVES: Students will be able to:

- trace a program that has no branching.
- trace a program that has unconditional branching.
- identify the value of each assigned variable at any step in a given program.
- recognize the effect of a GOTO statement in a program.

VOCABULARY: NEW: trace, GOTO, branching

REVIEW: line number, GOTO, output, variables, INPUT

MOTIVATION:

- Determine the output at lines 30 and 50 (if the user inputs the value of B as 12).

```
10 INPUT B
20 LET A = B
30 PRINT A, B
40 LET B = B + 5
50 PRINT A, B
60 END
```

- How do we follow the logic of the program to determine the output? (We should be able to test accuracy by a hand trace.)

AIM: How do we trace a program?

DEVELOPMENT:

1. Let's see what happens in each line of the program we just discussed.
2. Let's create a chart to trace what happens.

Values of Variables

Line No	A	B	Output
10	12		
20	12		
30			12, 12
40	17		
50			12, 17

3. Verify the accuracy of this trace by typing and running the program.

4. To handle more than one set of data, replace Line 60 with 60 GOTO 10.

    - This is called an unconditional branch. When prompt appears, type the following for the values of B:

        - 12
        - 5
        - 99

5. How can we trace a program that has an unconditional branch? (We go back to the beginning and trace the program for each new value.)

6. Let's continue the table:

Line No	A	B	Output
10		5	
20	5		
30			5, 5
40		10	
50			5, 10
10		99	
20	99		
30			99, 99
40	104		
50			99, 104

APPLICATIONS:

- For the programs below, do the following:
  - Explain what the progam will do.
  - Trace the program.
  - Create a trace chart.
  - Use 2, 10, 5, 25, 4, 5 for S (side).

```
10 INPUT S
20 LET A = S * S
30 PRINT "area is "; A
40 GOTO 10
```

  - Use 100, 300, 450 for M (miles).
  - Use 10, 15, 10 for G (gallons).

```
10 INPUT M
20 INPUT G
30 LET MG = M/G
40 Print "Miles per gallon are "; MG
50 GOTO 10
```

SUMMARY: Explain how a programmer checks the accuracy of his program.

EXERCISES (ASSIGNMENT):

- Write a flowchart for the "miles per gallon" program given in this lesson.
- Write flowcharts and prepare trace tables for each of the following programs.
  - <u>Batting Average</u>

```
10 INPUT H
20 INPUT AB
30 LET AV = H/AB
40 PRINT "Batting Average is"; AV
50 GOTO 10
```

  - Use 3, 33, 50 for "H"
  - Use 4, 100, 200, for "AB"

- <u>Average Points Per Game</u>

```
10 INPUT G1
20 INPUT G2
30 INPUT G3
40 LET S = G1 + G2 + G3
50 LET PG = S/3
60 PRINT "Points Per Game Are "; PG
70 GOTO 10
```

- Use following

G1	G2	G3
8	13	12
23	27	37

Lesson 25

PROGRAM TRACE WITH CONDITIONAL BRANCHING (IF/THEN)

PERFORMANCE OBJECTIVES: Students will be able to:

- test particular values of variables as they are processed through a computer program.
- identify the output produced by a program involving conditional statements.
- trace the mechanism of a flag.
- write a trace of a program involving conditionals.

VOCABULARY: REVIEW: conditional branching, trace table, flag

MOTIVATION:

- Predict the output of this program:

```
10 X = 11
20 X = X - 1
30 PRINT X
40 IF X > 0 THEN 20
50 PRINT "BLAST OFF"
60 END
```

- How is this program different from the ones we have traced before? (This program has conditioned branching.)
- How does conditional branching affect what happens in a program?
- Why must we keep track of the variable in this program?

AIM: How do we trace programs with conditional branching?

DEVELOPMENT:

1. Develop the following trace table:

LINE #	VALUE OF VARIABLE	TEST	YES OR NO	OUTPUT
10	11			
20	10			
30				10
40		10 > 0?	Yes	
20	9			
30				9
40		9 > 0?	Yes	
20	8			
30				8
40		8 > 0?	Yes	

- Continue to develop trace table for values 7, 6, 5, 4, and 3.

20	2			
30				2
40		2 > 0?	Yes	
20	1			
30				1
40		1 > 0?	Yes	
20	0			
30				0
40		0 > 0?	No	
50				Blast Off
60 END				

2. The following program includes a flag.
Determine the output.
Write a trace table.

```
10  READ N$, AV
15  IF N$ = "XXX" THEN 70
20  IF AV >= 90 THEN 40
30  GOTO 10
40  PRINT N$; " IS AN HONOR STUDENT."
50  GOTO 10
60  DATA JOE, 85, SUE, 95, TOM, 90, XXX, 1
70  END
```

LINE #	VALUE OF VARIABLE	TEST	YES OR NO	OUTPUT
10	Joe, 85			
15		JOE=XXX?	No	
20		85 >= 90?	No	
10	Sue, 95			
15		SUE=XXX?	No	
20		95 >= 90?	Yes	
40				Sue is an Honor Student
10	Tom, 90			
15		TOM=XXX?	No	
20		90 >= 90?	Yes	
40				Tom is an Honor Student
10	XXX			
15		XXX=XXX?	Yes	
70 END				

APPLICATION:

- Your class has taken two tests so far. Three students want to know if they are passing.
  - Write a flowchart to solve the problem.
  - From the flowchart, write a program. Use READ/DATA and a flag.
  - Write a trace table to verify the program.

  Sample program:

```
 10  READ N$, M1, M2
 20  IF N$ = "XXX" THEN 110
 30  AV = (M1 + M2)/2
 40  IF AV  = 65 THEN 70
 50  PRINT N$ "IS NOT PASSING"
 60  GOTO 10
 70  PRINT N$ "IS PASSING"
 80  GOTO 10
 90  DATA MARY, 65, 70, ROY, 55, 65,
100  DATA ALICE, 70, 75, XXX, 1, 1
110  END
```

SUMMARY:

How are the trace tables we did today different from trace tables previously learned?

EXERCISES (ASSIGNMENT):

- Trace the following program and determine the output. Use a trace table with appropriate headings:

```
10  LET N = 0
20  LET N = N + 1
30  LET P = N ↑ 2
40  IF P < = 100 THEN 20
50  PRINT N
60  END
```

- Write a program to print the names and grades of students who are in grade 10 only. Use a flag. Write a trace table.

Name	Grade
Sue	9
Tim	10
Joe	11
Pat	9

Lesson 26

COMPARISON OF STRING VARIABLES

PERFORMANCE OBJECTIVES: Students will be able to:

- Correct TYPE MISMATCH errors.
- Explain programs that make use of relational operators applied to character strings.
- Write a program that compares string variables.

VOCABULARY: NEW: sort, type mismatch, relational operators.

REVIEW: alphanumeric, string variables.

MOTIVATION:

- Arrange the list of words in alphabetical order:

  trust
  treat
  treaty
  trustworthy
  treatise
  tumult
  too
  taste

- For what purpose do we use alphabetized lists? (Dictionaries, telephone books, etc.)
- For what purpose do we use lists that are in numerical order? (Zip codes, batting averages, etc.)
- What do we call the process of putting things in alphabetical or numerical order? (Sorting)

AIM: How does the computer sort alphabetic data?

DEVELOPMENT:

1. How does the computer compare numeric data?

   $5 < 7$, $6 <= 6$, $10 > 1$, $7 <> 5$.

2. How does the computer handle alphanumeric data? (The computer assigns values to the strings and compares these values.)

   - treaty < trust, too > taste, tumult <> trustworthy, and treat < treaty.

3. Relational operators may be applied to string variables. What are the relational operators? (< , > , <=, >=, = , <>.)

4. Let's consider a program that accepts two character strings and outputs them in alphabetical order.

```
10  INPUT "FIRST STRING"; A$
20  INPUT "SECOND STRING"; B$
30  IF A$ < B$ THEN 70
40  PRINT B$
50  PRINT A$
60  GOTO 10
70  PRINT A$
80  PRINT B$
90  GOTO 10
```

5. RUN this program. Use Mary for A$ and Joe for B$.

6. RUN this program again. Use your name and the name of the person to your right.

7. How does the sorting capability of the computer work?

   - Note: Upper and lower case characters have different numerical equivalent values within the computer. (Generally upper case < lower case).

8. Type 10 A$ = 50.

   - What happened? (ERROR MESSAGE: TYPE MISMATCH appeared.)

   - What does this mean? (A string variable must be reserved for alphanumeric data.)

APPLICATIONS:

- Decide whether a "TYPE MISMATCH" error occurs in each of the following. Correct these errors.
    - 10 A$ = 500
    - 50 Y = "HELLO"
    - 100 Z$ = "10455"
    - 250 IF A$ = NO THEN 15
- Write a program to print the names of two friends in correct order. Use READ/DATA and a flag.

SUMMARY: How does the computer sort alphanumeric data?

EXERCISES: (ASSIGNMENT)

- Answer TRUE or FALSE for the following string comparisons:
    - APPLE < TOMATO
    - TEST > WAR
    - ATOM >= BASKET
    - TOM > TOMATO
    - SCHOOL <> HOME
    - 123 > 231
    - TOM >= MOT
    - OTTO <> TOOT
    - HOMEWORK < HOME
    - ETON <= NOTE
- Write a program that INPUTS the names of two months of the year and puts them in alphabetical order.

Lesson 27

CONDITIONAL STATEMENTS INVOLVING STRING VARIABLES

PERFORMANCE OBJECTIVES: Students will be able to:

- define relational operators with respect to string variables.
- write a program using conditionals and string variables.
- point out common errors in conditional programs involving string variables.
- trace a program involving conditionals and string variables.

VOCABULARY: REVIEW: string variable, relational operators, conditional statements,

MOTIVATION:

- Let's write a program that will ask students the multiple choice question, who is President of the USA?

```
10    PRINT "WHO IS PRESIDENT OF THE USA?"
20    PRINT "1. NIXON  2. FORD  3. CARTER  4. REAGAN"
30    INPUT "THE NUMBER OF THE CORRECT CHOICE IS"; N
40    IF N = 4 THEN 70
50    PRINT "INCORRECT, TRY AGAIN."
60    GOTO 10
70    PRINT "CORRECT."
80    END
```

- If we wanted to create a completion question to ask students who the President of the USA is, how would we write this program?

```
10    PRINT "WHO IS PRESIDENT OF THE USA?"
20    INPUT N$
```

- If we wanted the program to tell the student if the answer is right or wrong, what would we add?

```
30    IF N$ = "REAGAN" THEN 60
40    PRINT "INCORRECT. TRY AGAIN."
50    GOTO 10
60    PRINT "CORRECT."
70    END
```

AIM: How do we write programs with conditionals and string variables?

DEVELOPMENT:

1. How did we use the relational operator = with string variables?

   - IF N$ = "REAGAN"
   - NOTE: Use quotation marks

2. How have we used other relational operators with string variables? (If one string is before another alphabetically, we use < .)

   - Write a short program to place Peter and Paul in alphabetical order.

```
10    A$ = PETER
20    B$ = PAUL
30    IF A$ < B$ THEN 70
40    PRINT B$
50    PRINT A$
60    GOTO 90
70    PRINT A$
80    PRINT B$
90    END
```

3. Write the flowchart that describes this program.

4. Write a program trace to verify this program.

5. Write a program that will read a set of names. If the name begins with the letters A through J, print the message "Your appointment is Monday." Otherwise print "Your appointment is Tuesday." Use the names Wilson, Smith, Cook, Andrews, Jones.

```
 10    READ N$
 20    IF N$ = "XXX" THEN 100
 30    IF N$ < "K"
 40    PRINT "MR. "; N$; " YOUR APPOINTMENT IS
       TUESDAY."
 50    GOTO 10
 60    PRINT "MR. "; N$; " YOUR APPOINTMENT IS MONDAY."
 70    GOTO 10
 80    DATA WILSON, SMITH, COOK, ANDREWS,
 90    DATA JONES, XXX
100    END
```

APPLICATION:

- You want a class list where you INPUT ten students' names and put them into two groups, A-L and M-Z.
  - Write the flowchart.
  - Write the program.
  - Write a trace table.

SUMMARY:

- Choose three strings for W$, X$, Y$, and Z$ to make the following true:
  - X$ = Y$
  - X$ < Z$
  - W$ > Y$
- What is the correct format for a conditional involving a string?

EXERCISES (ASSIGNMENT):

- Find and correct any errors in the following statements:

```
10    IF "APPLE" = A$ THEN 25
20    READ X, Y, Z$
30    DATA "RIGHT", 10, 20
65    IF A$ = AIM THEN 50
75    IF "B$" < "COW" THEN 80
```

- True or False: IF N$ = "ABC DEF GH"
  - IF R$ = "ABC DEF GH" THEN R$ = N$
  - IF S$ = "ABCDEFGH" THEN S$ = N$
  - IF T$ = "BCDEFGH" THEN T$ < N$
  - IF W$ = "ABC DEF GH" THEN W$ = N$
- Write a program that will compute 8% sales tax on any amount that is inputted. Have a line in the program where the user can answer "yes" or "no" in order to have the sales tax program run again.
- Write a program that will read the words CLOUD, CHAIR, CROSS, COAT, CUTE and print only those words from C to CO alphabetically.

Lesson 28

COUNTING WITH IF/THEN

PERFORMANCE OBJECTIVES: Students will be able to:

- state the difference between incrementing and decrementing.
- list the steps in any counting loop.
- list at least three practical applications of counting.
- trace a given counting program.
- modify a given program to count from an initial number to a terminal number by a given increment.

VOCABULARY: NEW: loop, initialize, increment, decrement, terminal

REVIEW: counting

MOTIVATION:

- What is the output of this program?

```
10    K = 11
20    K = K-1
30    PRINT K
40    IF K< 0  THEN 60
50    GOTO 20
60    PRINT "BLAST-OFF"
70    END
```

- What practical applications are involved in having the computer count (forward or backwards)? (Time remaining in a sporting event, the number of employees processed in a payroll, the number of volumes in a school library all involve the idea of counting.)

AIM: How do we use the computer to count?

DEVELOPMENT:

1. How could we write a program to count from 1 to 10?

```
10     J = 1
20     PRINT J
30     J = J + 1
40     IF J > 10 THEN 60
50     GOTO 20
60     END
```

2. What is the purpose of line 10? (Initialization: We initialize [set] the value of the variable to zero.)

3. What is the purpose of line 30? (Incrementation: We increment (increase) the value of the variable by 1 for each pass through the program.)

4. In the count-down program, what is the purpose of line 30? (Decrementation: We decrease the value of the variable by 1 for each pass through the program.)

5. What is the purpose of line 40 in either program? (Stopping mechanism: When the value of the variable equals the limit, the program ends.)

6. Write a trace table for this program.

7. Let's modify the program to count from:
   1 to 6   (die)
   1 to 52   (cards)

8. How do we get the computer to count from any given number as an initial value, for example, 5 to 10 or 15 to 52?

9. Let's modify the program to count from 2 to 100 by twos.

   - What is the initial value? The final value? What change must be made in the incrementation step?

APPLICATIONS:

- Write a program to count from the initial value to the final value by the number indicated. Display all numbers.
  - 5 to 100 by 5's
  - 7 to 301 by 6's
  - 20 to 4 by 2's

- How do we count backwards on the computer? (Count from the highest to lowest and decrement by the number indicated.)

  - Example

```
10    X = 5
20    IF C > 100 THEN 60
30    PRINT X
40    X = X + 5
50    GOTO 20
60    END
```

- Rewrite this program to count backwards.

- Combine both programs so that the computer counts from 5 to 100 and then back to 5 again.

SUMMARY:

- What are the three steps involved in any counting loop?

- How does each step work?

EXERCISES: (ASSIGNMENT):

- Determine the output of this program:

```
10 K = 9
20 PRINT K
30 IF K > 156 THEN 60
40 K = K + 3
50 GOTO 20
60 END
```

- Rewrite this program to count backwards.

- Combine both programs so that it looks like the computer is counting forwards and then backwards.

- Determine the output of this program:

```
10 M = 20
20 PRINT M
30 IF M < 11 THEN 60
40 M = M - 3
50 GO TO 20
60 END
```

- Write a trace table to verify this program.

- Write the programs that will output these sequences of numbers:

    - 2 to 48 by 2's
    - 5 to 455 by 9's
    - 100 to 50 by 2's
    - 1.5 to 152.5 by 1's

Lesson 29

INTRODUCTION TO THE FOR/NEXT LOOP

PERFORMANCE OBJECTIVES: Students will be able to:

- define the function of a FOR/NEXT loop.
- use the FOR/NEXT loop in a simple program.

VOCABULARY: NEW: FOR/NEXT
REVIEW: GOTO, IF/THEN, loop, READ/DATA

MOTIVATION:

- Write a program using only one PRINT statement to display the following phrase ten times on the screen: "I AM A COMPUTER."
- How can we print something ten times when we are allowed to use only one print statement? (We go back to the same print statement ten times.)
- What do we call it when we want the same instruction to be followed repeatedly? (A loop.)
- What are the different types of loops we have learned? (Unconditional [GOTO] loop and conditional [IF/THEN] loop.)
- In the program you were just asked to write, why must you use the IF/THEN loop?
  - Sample program:

```
10 N = 1
20 PRINT "I AM A COMPUTER"
30 N = N + 1
40 IF N> 10 THEN END
50 GOTO 20
```

- Why do we send the program (line 50) back to line 20 and not to line 10? (Returning to line 10 would re-initialize the variable (N) to 1 and we would never reach ten.
- Why is it important to include line 10? (Line 10 initializes N to the value we wish.)

- What is the function of line 30? (Line 30 increases the value of N by 1 each time and acts as a counter.)
- Today we will learn another way to write a program to do something a certain number of times. This is another conditional loop called FOR/NEXT.

AIM: How can we use the FOR/NEXT loop to make the computer execute the same set of instructions more than once?

DEVELOPMENT:

1. We'd like the computer to perform some act FOR a given number of times. To have it do this, we will use a FOR/NEXT loop.
   - The name actually reflects the function: it tells the computer to do something for a specified number of times. The computer then repeats anything placed between the words FOR and NEXT the requested number of times.

   ```
   10 FOR N = 1 TO 10
   20 operation to be performed
   30 NEXT N
   40 END
   ```

2. What does line 10 do? (Line 10 tells the computer that an operation must be performed ten times.)
3. What does line 20 do? (Line 20 specifies the operation to be performed.)
4. What does line 30 do? (Line 30 tells the computer to go back to line 10, check off that the operation has been performed, and then perform it again.)
5. How does the computer know when to stop going through this loop? (Line 10 told it to perform the operation ten times. After the tenth time, the program will drop to line 40.)
6. Why must the variable in line 10 be a numeric, not a string variable? (N acts as a counter so it needs a number.)
7. Change one line of the program so that the operation is performed 15 times.
   - 10 FOR N = 1 TO 15

8. Rewrite the sample program used in the motivation, this time using a FOR/NEXT loop instead of the IF/THEN statement. (This can be accomplished by inserting PRINT "I AM A COMPUTER" in the above example as line 20.)

9. In a FOR/NEXT loop, why don't we initialize our variable as in the IF/THEN loop? (In a FOR/NEXT loop the computer automatically initializes the variable to the starting value.)

10. Predict the outcome of the following program:

```
10 PRINT "NAME", "AGE"
20 FOR N = 1 TO 3
30 READ N$, A
40 PRINT N$, A
50 NEXT N
60 PRINT "TOTAL IS 3 STUDENTS"
70 DATA "MARY",15,"GEORGE",14,"JOHN",16
80 END
```

11. What is the first value that N will be set at? (One.)

- What line in this program starts the loop? (Line 20.)
- What is this program asking the computer to do? (In line 30, the computer will be told to read the first two items ,the string variable N$ and the numeric variable A. from the DATA statement in line 70. Line 40 will print those values on the screen under the headings set up in line 10.)
- How many times will the computer execute this loop? (Three times.)
- What does the computer do when it reaches line 50? (It returns to line 20 and adds one to the value of N.)
- When will the computer end this loop (After N becomes greater than 3.)
- What will the computer do after it finishes the loop? (It will go to the next line--in this case line 60--and print "TOTAL IS 3 STUDENTS"; the program will then end at line 80.)

12. What is the function of the following program lines?

```
35 FOR X = 1 TO 50
38 NEXT X
```

- This is called time-delay. The computer counts to itself from 1 to 50.

- Insert these lines into the program above and RUN the program.

- What happened? (The program was slower.)

APPLICATION:

- Predict the outcome of each of the following examples:

  -
    ```
    10 A$= "I CAN'T STOP REPEATING
       MYSELF"
    20 FOR N = 1 TO 100
    30 PRINT N, A$
    40 NEXT N
    50 PRINT
    60 PRINT "FINALLY!!!"
    70 END
    ```

  -
    ```
    10 PRINT "#", "2X#", "3X#"
    20 FOR N = 1 TO 20
    30 PRINT N, 2*N, 3*N
    40 NEXT N
    50 END
    ```

- Write a program to print seven asterisks in a column.

- What would be the effect of adding a time-delay?

SUMMARY:

- What three statements can make a program loop? (GOTO, IF/THEN, FOR/NEXT.)

- Write programs to print the word "loop" ten times. Use a GOTO loop, an IF/THEN loop, and a FOR/NEXT loop.

  -
    ```
    10 PRINT "LOOP",
    20 GOTO 10
    ```

  -
    ```
    10 N=0
    20 IF N=10 THEN END
    30 PRINT "LOOP",
    40 N=N + 1
    50 GOTO 20
    ```

```
. 10 FOR N=1 TO 10
  20 PRINT "LOOP",
  30 NEXT N
  40 END
```

- What are the similarities?

- What are the differences?

EXERCISES (ASSIGNMENT):

1. Write the outcome of each of the following examples:

```
. 10 A$ = "HAPPY "
  20 B$ = "BIRTHDAY  "
  30 FOR N = 1 TO 20
  40 PRINT A$;B$
  50 NEXT N
  60 END
```

```
. 10 FOR N = 1 TO 1000
  20 PRINT 3*(N+2)
  30 NEXT N
  40 END
```

```
. 10 FOR N =  1 TO 2
  20 READ A$,B$
  30 PRINT A$;B$
  40 NEXT N
  50 DATA ABE, LINCOLN
  60 DATA GEORGE, WASHINGTON
  70 DATA BEN, FRANKLIN
  (Poor Ben's name will be left
   out in this example.)
```

```
. 10 FOR N = 1 TO 5
  20 PRINT N " SQUARED IS";N*N
  30 NEXT N
  40 END
```

2. Write a program to print a list of students' names and test scores. Use FOR/NEXT, READ/DATA, and a time-delay. The information is in the following chart:

Student	Test 1	Test 2	Test 3
Anna	75	70	80
Daniel	65	80	75
Mary	55	65	65
Jack	80	90	80

- Add headings and blank lines for a nice looking output. Add REM statments to explain what is happening in your program.

Lesson 30

COUNTING, TOTALLING, AND INPUTTING USING A FOR/NEXT LOOP

PERFORMANCE OBJECTIVES: Students will be able to:

- predict the outcome of a program containing a FOR/NEXT loop.
- use the FOR/NEXT loop in a program allowing user INPUT.
- use the FOR/NEXT loop in a simple counting program.
- use the FOR/NEXT loop in a more sophisticated program capable of producing a running total.

VOCABULARY:	NEW:	accumulator
	REVIEW:	FOR/NEXT loop, READ/DATA, INPUT

MOTIVATION:

- Write a program using a FOR/NEXT loop to count from ten to twenty.

```
20 FOR N = 10 TO 20
30 PRINT N
40 NEXT N
```

- Explain the format for a FOR/NEXT loop.

```
10 FOR (numeric variable) = (starting number)TO
   (ending number)
20 operation
30 NEXT (numberic variable)
```

- What happens if we omit the NEXT statement? (The computer gives us a FOR WITHOUT NEXT error message.)
- For what purposes have we used FOR/NEXT loops? (Counting programs and time-delay.)
- What is the proper format for a FOR/NEXT loop? (The first line reads FOR (numeric variable) = (starting number) to (ending number).) The next line(s) contains the operation to be repeated. The last line must contain the statement NEXT (numeric variable).

- What would have happened if we omitted the NEXT statement in line 40? (The computer would have given us a "FOR WITHOUT NEXT" error.)

- What must the starting value for N be? (Ten, since we must count from ten to twenty)

- Do we have to "initialize" our variable to "LET" it equal a certain number? (No, not in a FOR/NEXT loop; the variable is automatically assigned the starting value.)

- It seems clear that the FOR/NEXT loop is very useful in writing simple counting programs. Now let's see if we can write more complex programs using this versatile loop.

AIM: How can we use the FOR/NEXT loop to write more sophisticated programs?

DEVELOPMENT:

1. . How can we keep a running sum of numbers in a program? (We must have another variable which will represent the sum of all the numbers here, the sum of the numbers 10 to 20.)

   - What variable could we use to accomplish this? (We can use any numeric variable that we haven't used before in this program.)

   - If we use the variable "S" to represent the sum of the numbers, what equation can we write to total all the numbers? (S = S + N)

   - What does each part of this equation mean? (The new sum is equal to the old sum plus the value of the number N.)

   - Where in the program should we include this statement? (It should be included within the FOR/NEXT loop.)

   - 25 S = S + N

2. Since S is not a counter within the FOR/NEXT loop, it (unlike N) must be initialized.

   - What should the initial value of "S" be? (Zero, because nothing has been added into the sum.)

     - Where in the program should we place the initial value of S? (Before the loop: 15 LET S = 0.)

BEST COPY AVAILABLE

- What would happen if this line were within the loop? (Every time the program reached that line, it would set S back to zero instead of increasing it.)

- What statement should we include in this program to allow us to see the sum? (We must include a PRINT statement in order to see the results.)

```
15 S = 0
20 FOR N = 10 to 20
25 S = S + N
30 PRINT N, S
40 NEXT N
```

3. RUN this program.

   - What were the results? (Two columns of numbers.)

   - How can we add titles to the screen display to clarify what these 2 columns of numbers represent? (Add the line PRINT "NUMBER", "SUM".)

   - Where should we put this line in our program so it will place the titles at the top of each column? (Since we want the titles to appear only once on the screen, we must place the line outside of the loop and before we print any of our results.

   - Include this line (18 PRINT "NUMBER", "SUM") and rerun the program.

4. How could we change this program to allow the person running it to start and end counting at any number of his or her choice? (We must include an INPUT statement to ask the user for the beginning number and ending number.)

   - What are some possible prompts and statements that would allow the program to accept two numbers from the user. (5 INPUT "ENTER TWO NUMBERS AND I WILL COUNT AND ADD THEM FOR YOU ";A,B)

   - How must we change the FOR/NEXT statement to allow it to use our inputted numbers? (Line 20 should now read FOR N = A TO B) Type in the new lines (5 and the revised 20) and run the program. Vary the values of A and B.

5. If a user runs this program, what might he do so that the program won't perform as expected? (If the first value entered is larger than the second value, the program won't work.)

- How can we have the computer check the inputted values before executing the FOR/NEXT loop? (Include a line after the INPUT to test if the first value is larger than the second. If it is, the computer could print a message to tell the user of the problem and then ask him again to input the numbers.)

```
 5  INPUT "ENTER TWO NUMBERS AND I WILL COUNT FROM
    ONE TO THE OTHER AND ADD THEM FOR YOU ";A,B
10  IF A > B THEN PRINT "THE FIRST NUMBER MUST BE LESS
    THAN THE SECOND NUMBER. ":GOTO 5
15  S = 0
18  PRINT "NUMBER", "SUM"
20  FOR N = A TO B
25  S = S + N
30  PRINT N, S
40  NEXT N
```

6. What can we add to make this program more sophisticated?

    - Clear/Screen function
    - REM Statements
    - Underlining and blank lines

  - Add these items to your program

APPLICATION:

- Determine the output of each example:

  -
    ```
    10 S = 12
    20 FOR N = 1 TO 8
    30 PRINT S
    40 S = S + 4
    50 NEXT N
    ```

    (This program prints 8 numbers in the sequence 12,16,20...)

  -
    ```
    10 INPUT "ENTER 2 NUMBERS";A,B
    20 FOR N = 1 TO 9
    30 PRINT A
    40 A = A + B
    50 NEXT N
    ```

    (This program prints the first nine numbers in a sequence starting at number A, and increases the value by B.)

- Write a program using a FOR/NEXT loop to produce a chart that would be helpful to a store to calculate an 8.25% sales tax on amounts ranging from 1 dollar to 20 dollars. Write a flowchart first.

```
.  5 PRINT "AMOUNT", "SALES TAX"
  10 FOR N = 1 TO 20
  20 P = .0825 * N
  30 PRINT N,P
  40 NEXT N
  50 END
```

2. Change the program to ask a person at what amount he would like his sales tax chart to start and at what amount he would like it to end. Include prompts for the user.

3. Write a program using a FOR/NEXT loop to input a person's first name and print that name six times.

    - Now write a program to accomplish the same task, but instead of writing the program using a FOR/NEXT loop, use an IF/THEN loop.

SUMMARY:

- When would we want a FOR/NEXT loop in a program? (When we want to repeat the same procedure such as counting and totalling a certain number of times.)

- What statement could we include in a counting program so its output could be changed by the user without having to retype any of the program lines? (We could use an INPUT statement to allow the user to vary the starting and ending values in the FOR/NEXT loop.)

- What are some important things to remember when using an INPUT statement with a FOR/NEXT loop?

- Include enough information in the prompt to allow the user to run the program.

- Anticipate any mistakes the user might make.

- Check the user's INPUT to see if it is correct before continuing the program.

EXERCISE (ASSIGNMENT):

- Determine the output of each example:

  -
    ```
    10 FOR N = 2 TO 5
    20 S = S*N - 3
    30 PRINT N,S
    40 NEXT N
    ```

  -
    ```
    10 A = 7
    20 FOR N = 1 TO 10
    30 READ B
    40 IF A <B THEN A = B
    50 NEXT N
    60 PRINT A
    70 DATA 5, 8, 3, 8, 2, 5, 1, 6, 3, 7
    ```

- Write a program to display the 7th through 15th number in the following progression: 3,9,12,15, etc.

Lesson 31

INTEGER AND RANDOM NUMBER FUNCTIONS

PERFORMANCE OBJECTIVES: Students will be able to:

- define and explain the purpose of the terms RANDOM and INTEGER.
- use the RND and INT functions in a program.

VOCABULARY: NEW: INTEGER (INT), RANDOM (RND)
REVIEW: FOR/NEXT, INPUT

MOTIVATION:

- Write a program which asks the user to guess a number between 1 and 10. Let's be nice; if the person doesn't guess right on the first try, tell him or her whether the guess was too high or too low and allow for another try.

```
10  INPUT "I'M THINKING OF A NUMBER BETWEEN 1 AND 10.
    TYPE YOUR GUESS FOR THE NUMBER ";A
20  N = 7
30  IF A=N THEN PRINT "YOU ARE CORRECT":END
40  IF A<N THEN PRINT "TOO LOW. TRY AGAIN":GOTO 10
50  PRINT "TOO HIGH. GUESS  AGAIN": GOTO 10
```

- How long might the user want to play this guessing game program as it is? (Not too long, since the user will get the number after a few tries and he or she will then know it. Therefore, it wouldn't really be a guessing game the second time he or she ran the program.)
- What would we have to do in order to change the hidden number? (We have to change the value of N in line 20.)
- What would be the best solution? (If the computer could determine the secret number by itself.)
- Let's see if there is a way to allow the computer to pick its own number.

AIM: How can a computer be programmed to pick a number at random?

DEVELOPMENT:

1. What is the advantage of working with whole numbers instead of numbers with decimal places? (It is easier for the user to understand.)

2. What word means whole numbers? (Integer)

   - What is the BASIC designation? (INT)

   - What exactly is an integer? (An integer is any whole number, positive, negative, or zero.)

3. Let's use the INT function:

   - Type in this program:

```
10  INPUT N
20  R = INT (N)
30  PRINT N,R
40  GOTO 10
```

   - RUN this program using the following for the value of N:

```
 10
  0
 -5
 3.5
-3.5
```

   - Explain the output in each case:

```
10, 10
 0,  0
-5, -5
3.5, 3
-3.5, -4
```

4. In the last example, why did -4 appear? (-4 is smaller than -3.5.)

   - Predict the output for the following:

```
N = INT (7.2)
N = INT (-8.9)
N = INT (7.7)
```

- Answers:
  7.2, 7
  -8.9, -9
  7.7, 7

5. What function do we use if we want the computer to pick a number at random instead of waiting for our INPUT? (The RANDOM function tells the computer to pick a number.)

   - What is the BASIC designation? (RND)

6. Type this program:

```
10  N = RND (0) or
10  N = RND (1) (This depends on the computer.)
20  PRINT N
```

   - What is the result? (The computer picks a number (N) between zero and one.)

7. How can we get the computer to pick a whole number? (Use the INT function.)

   - Type

```
10  N = INT (RND(0)) or
10  N = INT (RND(1))
20  PRINT N
```

   - What happens? (The computer always prints zero because with the INT function, the computer uses the number before the decimal point. It doesn't round off.)

8. What operation could we perform before using the INT function to widen the range of numbers? (We can multiply the decimal produced by RND by a number.)

   - How should this look?

```
N  = INT (X*RND(0)) or
N  = INT (X*RND(1))
```

   - How do we write the value for X? (We use a LET statement or insert the value for X in the program line.)

   - How do we choose the value for X? (The greater the value of X, the greater a range of numbers the computer has to choose from.)

9. How do we get the computer to choose a number from one to ten?

```
N  = INT (10*RND(0)+1) or
N  = INT (10*RND(1)+1)
```

- How do we get the computer to choose a number from one to six?

```
N  = INT (6*RND(0)+1) or
N  = INT (6*RND(1)+1)
```

APPLICATION:

- Match column I with its correct answer in column II. (Note that one answer from II is used twice.)

I

1. RND (1)
2. INT(2.2)
3. RND (0)
4. INT (4*RND(1))
5. INT(RND(1))
6. INT(-7.9)
7. INT(4*RND(0)+1)

II

a. 0
b. numbers from 1 to 4
c. 2
d. -8
e. numbers greater than 0 but less than 1
f. numbers from 0 to 3

- Write a program to print 25 numbers chosen at RANDOM in the range of 1 to 10.

```
10 FOR N = 1 TO 25
20 X = INT(10*RND(0)+1)
30 PRINT N
40 NEXT N
```

- Write a program to simulate rolling a die twenty times and print the number of times the die comes up five. Remember, a die has six sides, so we could roll any number from one to six.

SUMMARY:

- What is the most useful form of the RND function for generating random whole numbers? (N=INT(X*RND(0)+1))

- What does each part of that equation do? (The X tells you how many numbers to choose from and is also the highest number; the one tells you that the lowest number will be one.)

- What happens when you apply the INT function to a positive number? (The computer prints only the whole number before the decimal point.) What happens when you apply the INT function to a negative number? (The computer prints the next lowest number, e.g., INT(-2.1) = -3.)

EXERCISES (ASSIGNMENT):

- We have two dice and we roll them at the same time. Write a program to determine how many times we will roll a lucky seven out of 25 rolls.
- Rewrite our guessing game program to allow the computer to select the number to be guessed.

Lesson 32

GENERATING RANDOM NUMBERS IN PROGRAMS

PERFORMANCE OBJECTIVES: Students will be able to:

- explain how INT and RND functions are used in a program.
- use the INT and RND functions in a program.
- write a flowchart demonstrating the use of the INT and RND functions in a program.

VOCABULARY: Review: INT, RND

MOTIVATION:

- If you were tossing a coin, what type of bet could you make on the outcome of the toss? (Heads or tails.)
- How can a computer flip a coin? (We could use the RND function to allow the computer to choose between the numbers 1 and 2 to determine if it tossed heads or tails.)
- What formula could we use to choose two random numbers, 1 and 2?
  - X = INT 2 * RND (0) + 1
- How would we know if it came up heads or tails?
  - IF X = 1 THEN PRINT "HEADS"
  - IF X = 2 THEN PRINT "TAILS"
- How do we get the computer to play this game? (Elicit AIM.)

AIM: How can the INT and RND functions be used in programming?

DEVELOPMENT:

1. Let's write a program to toss a coin. We'll ask the user if he wants to risk his money. If he does, we'll give him $25.00 to start and allow him to bet part or all of the money.

- What are some important things we must consider when allowing this person to bet?
  - We must check to see if the player has enough money to cover his bet. If he doesn't, we must tell him and allow him to try again.
  - We must decide if we will give him credit.

2. What must we ask the person about his or her bet? (Does he want heads or tails?)

3. How do we get the person to believe the computer's toss? (Show what the result is.)

4. After the coin is flipped, what remains to be done? (Decide if the person was right (IF/THEN) and tell him.)

5. What do we do with his bet? (Add or subtract the amount of the bet from his total.)

6. What can we tell him after that? (If he still has money, we can invite him to play again. If he is broke, we can give him credit to play again.)

7. Write a flowchart for the program for this game.

8. Before we write the program, what extra touches could we use?
   - REM statements.
   - Display the amount of his winnings.
   - Keep a running total of what the person bet.

9. Write this program.

APPLICATION:

- Play the game.
- Add any other program lines that will make it look professional.

SUMMARY:

- How do we combine INT and RND functions to generate random numbers from 1 to X?

- What are some good techniques to use when writing a program using the RND function?

EXERCISE (ASSIGNMENT):

- Write a program to play a game of dice. These are the rules:
    - If you roll a 2, 3, or 12, you lose automatically.
    - If you roll a 7 or 11, you win.
    - If you roll a 4, 5, 6, 8, 9, or 10, that number becomes your point and you must roll again. You continue to roll until you get your number again to win or an automatic win or loss.
- Write the flowchart for this program.

Lesson 33

THE ADVANTAGES OF COMPUTERS

PERFORMANCE OBJECTIVES: Students will be able to:

- explain how computer use can benefit individuals.
- identify how individuals may benefit from personal computing.
- explain how government and business can benefit from computer use.
- suggest possible future uses of computers.
- explain the types of tasks computers are used for.

VOCABULARY:	NEW:	UPC universal pricing code, retrieval, data processing, process control, simulation
	REVIEW:	data processing

MOTIVATION:

- List some electronic devices that are part of your life.
    - Examples:
        - digital watch
        - touch-tone phone
        - calculator
        - television with electronic tuning
        - microwave oven
        - cash register
- How are these things related to computers?
- How have electronic devices improved our lives?

- speed of obtaining results
- ability to accomplsh more work
- improved accuracy
- improved efficiency

AIM:

- What are the advantages of computers to individuals, business, and government?

DEVELOPMENT:

1. What are the tasks that computers perform?
    - Retrieval - getting information back that is already input: checking your balance in a bank account.
    - Processing - taking data and doing something with it: adding numbers together.
    - Process control - regulating certain situations: a thermostat checks the temperature and turns on heat or air conditioning.
    - Simulation - imitiating a real-life situation on the computer in order to find the best solution: Engineers in the space program simulate different courses for the space ship in order to find the best one.
2. What are the major differences between the different generations of computers? (Their physical size, their speed of operation, their memory size, and their cost.)
    - What improvements have there been? (Physical size is down, speed and memory are up and cost is down. Scientists are using computers to design new computers which are even smaller and faster.)
3. If a computer were small enough and inexpensive enough for you to own and bring to school, what would you use it for?
    - games
    - math drill
    - studying facts: social studies, dates, events

- spelling
- reinforcement of skills (reading, math, etc.)
- memorizing facts, poems, etc.
- learning geography (map skills)
- printing compositions without re-writing (word processor)
- studying grammar
- listing homework assignments
- listing grades
- etc.

4. How can your school use computers?

   - scheduling classes - printing student schedules
   - printing report cards
   - saving student grades
   - computing averages
   - scoring tests
   - checking daily attendance
   - checking class cutting
   - college selection
   - teaching programming
   - teaching and reinforcing subjects taken by students

5. How can you use computers at home?

   - Keeping family records: taxes, bank accounts, phone numbers and addresses, recipes, important dates.

   - Entertainment: video games.

   - Education: learning programming, studying what was learned in school, using educational programs.

6. Sometimes we add a telephone line hookup to a home computer. What would that do?

   - Communicate with stores, banks, and other services by using your monitor to browse through a store catalogue.

   - Use your keyboard to order and transfer money from your bank account to the store account.

7. What tasks in the future do you see computers used for that can affect you?

   - Do all shopping by computers.
   - Read electronic newspapers, weather reports.
   - Have the latest movie sent by cable to your T.V.
   - Take courses via computer.

8. How are computers used in the business world? (Payroll, inventory, employee records, research, marketing strategy, electronic mail, etc.) Businesses use computers to increase productivity and profits.

9. How are computers used in:

   - Supermarkets - Universal Pricing Code.
   - Factories - robots.
   - Banks - savings accounts, checking accounts, loan payments, calculating interest, credit history.
   - Utility companies - calculating how much utility is used, figuring the price, printing bills, monitoring payments, projecting future needs and peak periods.
   - Hospitals - intensive care units monitor body functions, heart rate, blood pressure, breathing temperature, etc., blood analysis, patient medical history.
   - Lending companies - credit history, checking employment, income, monitoring payments, figuring interest, deciding if you are a good risk.

10. For what other purposes do businesses use computers?

11. How does the government use computers?

    - The census.
    - Tax collection - Internal Revenue.
    - Secrets - military - espionage.
    - Defense - missiles, aircraft, ships, simulate attacks and defense plans.
    - Personal records of civilians and military personnel.
    - Train astronauts in the space programs, flight simulators, collect information from space voyages.

- Criminal records and investigation.
- Distribution of money such as tax refunds, social security payments, Medicare, etc.
- Local governments, car registration and license, traffic control, wanted persons, stolen property.
- Investigation: voice prints, lie detector, fingerprinting records.

APPLICATIONS:

- Explain the advantages of using computers in the following:
    - an employee in a fast food restaurant taking a large order.
    - an owner of a gas station.
    - a shopper in a department store waiting on line to pay.
    - investigators at the FBI.
    - you're making a deposit to your savings account.
    - your school's attendance office.
    - your teacher who has to mark 123 exams .
    - the space program.
    - a student practicing math examples.
    - weather forecasting.
    - a hospital patient who needs to have his breath analyzed for proper amounts of oxygen.
- Describe the type of activity the computer is used for in each of the above (retrieval processing, process control, simulation.)
- Are these future applications possible? Explain.
    - Weather control.
    - Sea mining.
    - Artificial life.
    - Control of heredity.
    - Space mining.
    - Immortality.
    - Others.

SUMMARY:

- What are the benefits of using computers?
- Who benefits from them?
- Describe what most computers do.
- How can we use computers in the future?

ASSIGNMENT:

- Are computers used for instruction in your school? If so, how? If not, what kind of instruction could the computer be used for?
- How do businesses improve productivity by using computers? Give some examples.
- How does government use computers to run a more efficient operation? Give examples.
- What other applications are there for computers? (In the area of health aids to the handicapped, research.)

Lesson 34

THE DISADVANTAGES OF COMPUTERS

PERFORMANCE OBJECTIVES: Students will be able to:

- explain how computers may be used directly against individuals.
- explain how computers can cause loss of jobs.
- discuss how computers can be used in committing crimes.
- discuss the feelings of depersonalization that computer use can cause.
- explain how we have become dependent on computer technology.

VOCABULARY: NEW: data bank, depersonalization, computer crime, "tourists," "crashers"

MOTIVATION:

- Give examples of situations when a computer was blamed for something that went wrong such as:
  - a cutting slip or absence note sent to your home in error.
  - a bill sent to you home after it was paid.
  - having to wait on line for long periods of time because the computer was "down."
- How often do computers make mistakes? (Computers rarely make errors. When a computer is blamed for an error it is most likely an error made by a person in programming, wrong information entered, or failure to enter all data.)
- Why do some people dislike computers?

AIM: What are the disadvantages of computer use?

DEVELOPMENT:

1. What numbers do we have that identify us at work, school, etc? (Social Security, school ID, insurance and health policy numbers, etc.)

    - Why do we need numbers? (Computers use numbers.)

    - Who gives us these numbers? (Government, corporations, credit agencies, schools, etc.)

    - What information can be stored under these numbers?

        - name
        - physical characteristics
        - employment
        - medical history
        - education
        - financial history
        - car registration
        - licences
        - police records
        - charge accounts
        - insurance
        - memberships

- What problems can this present about privacy and confidentiality?

    - A company can maintain hearsay evidence or unnecessary information about a person without the person having access to this information or the opportunity to correct mistakes or challenge misrepresentations.

    - Unfavorable entries can be retained too long. Computers make it possible for records of youthful mischief, delinquent payments, etc. to be held indefinitely.

    - Records about individuals can be exchanged freely among agencies. A prospective employer may be able to get high school records or a bank report or medical records, even though the employee may not know this is happening.

    - Governments, individuals or businesses can use information stored in computers as a tool for blackmail or intimidation. For example, Governor X is running for re-election. He wants Mr. Y to support him. By using information in a computer, Governor X learns that Mr. Y was convicted of tax evasion 15 years ago. Governor X threatens to publicize the conviction if Mr. Y doesn't support him.

2. How can the computer's ability to perform tasks faster than humans affect us? (Since computers can increase productivity for the company and thereby increase profits, some workers may lose their jobs.)

    - What types of jobs can be eliminated by computers? (File clerks, bookkeepers, etc.)

    - Although some jobs are eliminated by computers, new jobs are created. What are some of these jobs?

    - Data entry clerk
    - Programmers
    - Designers
    - Engineers
    - Technicians
    - Maintenance people

3. What is computer crime?

    - What are some examples?

        - With the proper equipment, people can spy on long distance telephone calls that are sent via satellite.

        - Messages sent from one terminal to another can be intercepted and used for stealing business secrets, research data, marketing plans, etc.

        - Data bank vandals and thieves use electronic bulletin boards that contain information on how to penetrate computer networks or the phone system.

        - "Tourists" are people who penetrate computer systems just to look without disturbing the data.

        - "Crashers" will erase or change files causing massive disruption to a business or government agency and, of course, to the individuals whose records are changed.

    - Certain individuals have gotten into business or banking systems by finding out passwords and codes. They have stolen by electronically transferring money from one business account to another account.

  - Who is affected by "computer crime"?

  - Who pays for it?

  - Give an example of a crime committed by using computers.

  - How are crimes committed with computers?

4. What is depersonalization? (The feeling that the organization a person is dealing with doesn't care or cares more about the information than about the person giving it.)

   - How can computer use cause the feeling of depersonalization to occur?

   - What experiences have you had that has lead to this feeling?

APPLICATION:

- Because of the many advantages we gain from computers, our society has become dependent on computers.

  - What would happen if computers were no longer available to the banking system, your school, the phone company, the United States missile defense system?

  - Debate the following scenario:

    - You pay a fee for a test given by an employment agency. You take the test. On another day, you go to apply for a job at a company. The company calls the employment agency to ask if they had any information on you. You find out later that you lost the job because of poor scores on the test. Did the agency have the right to give your test scores out to the company?

  - If you had access to school, health, welfare, police and psychiatric records and wanted to predict who possible law offenders might be, how would you do this?

- What do you think of the idea?

SUMMARY:

- What are some disadvantages we face because of computer use?
    - Dependence
    - Loss of privacy and individuality
    - Loss of certain jobs in the economy
    - Increased threat of crime
    - Possible loss of freedom
- What are examples of each?

EXERCISE:

- List the pros and cons of having a national data bank which would include the following:
    - Name
    - Social Security number
    - School history
    - Employment history
    - Credit history
    - Medical history
    - Organizations belonged to
    - Magazines subscribed to
    - Phone calls made
    - Insurance
    - Travel history
- "A decade ago most of us assumed that computers were the servants of mankind. Now we realize that computers may be used as weapons to assault our right to privacy." Ognibene, Peter, "The Keeper of Secrets" Omni, March, 1983. Write a composition in which you agree or disagree and explain your opinion.
- Does computer technology give an individual more or less freedom? Write an essay in which you state and explain your opinion.
- List ways in which you are dependent on computers.

Lesson 35

HISTORY OF COMPUTER LANGUAGES

PERFORMANCE OBJECTIVES: Students will be able to:

- identify the need for different types of computer languages.
- name at least six computer languages and tell what each is used for.
- compare the way several computer languages are used to solve the same problem.
- identify the characteristics of the main computer languages.

VOCABULARY: NEW: interpreter, compiler, machine language, programming languages, assembly languages, FORTRAN, COBOL, PL/I, BASIC, Pascal, RPG, LISP, ALGOL, APL, ADA, Logo.

MOTIVATION:

- What does the following sentence mean: "I drove a lorry yesterday."
- What is wrong with the statement above as a communication message? (The person to whom you are communicating may not understand the meaning of lorry.)

AIM: Why are computer languages necessary?

DEVELOPMENT:

1. In order to communicate with a computer, we must use a series of instructions called a program. The early computers had programs that consisted of wires attached to pins inside the computer. If you wanted to change the program, you had to change the wiring. Errors in a program were difficult to trace.

2. A programming language is a set of rules that can be used to provide instructions to a computer. Unlike the old hard-wired programs, today's computing languages provide instructions and a limited vocabulary, which make it easier to communicate with the computer. There are over 150 different computer languages in use today.

3. Low-level languages are the closest to the language used by a computer. Machine language uses the binary number system with 0's and 1's. It is difficult to learn and easy to make mistakes in machine code. Assembly languages use codes to replace binary digits. They use an assembler program to translate into machine language. Each type of computer has its own assembly language. They are very detailed. It is easy to make errors.

4. High level languages use English words and are easier to use than machine and assembly language. Some, like BASIC, use an interpreter which translates line-by-line into machine code. Interpreted languages speed up the writing of programs but are slower to execute. A compiler translates the whole program before executing it. Compiled languages take longer to write but are shorter in execution time.

5. FORTRAN (Formula Translator) was introduced by IBM in 1954. It is a scientific language, and is good for solving formulas but poor for file processing and data processing.

6. COBOL (Common Business-Oriented Language) was developed in 1959 and updated several times since then. COBOL is very similar from one type of computer to another. It is used in business.

7. BASIC (Beginners' All-Purpose Symbolic Instruction Code) was developed at Dartmouth in 1965. It is easy to use. Most microcomputers use a version of BASIC.

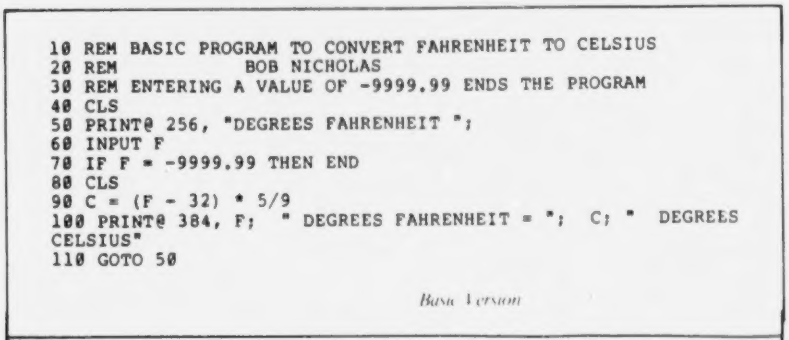
8. Pascal is a structured language developed in 1971 and used in science programs.

9. LOGO was developed for research purposes and adapted by Seymour Papert as an educational vehicle. LOGO users control a turtle to construct geometric shapes.

10. Other programming languages are PL/I, RPG, LISP, ALGOL, SNOBOL, and APL.

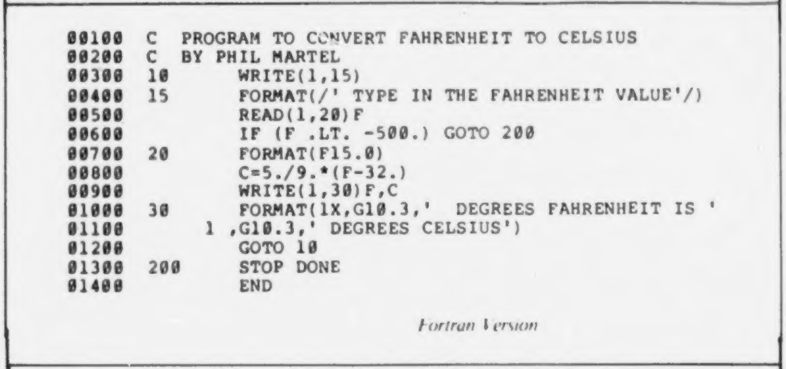
SUMMARY:

- See programs below (to be duplicated and distributed to students).
    - Which is the shortest program? Which is the longest?
    - In the BASIC program, what statement will end the program?
    - Which language uses the word "format"? What English words do you recognize in that program?
    - What three divisions are there in the COBOL version? In which division is the computation done?
    - Which program uses IF-THEN-ELSE?
    - Is the APL program easy to understand? Explain.



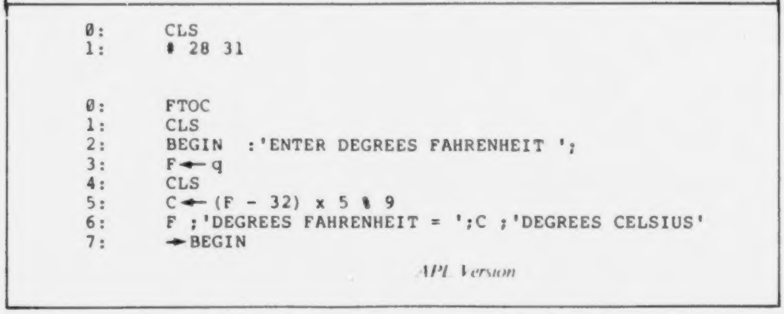
```
10 REM BASIC PROGRAM TO CONVERT FAHRENHEIT TO CELSIUS
20 REM             BOB NICHOLAS
30 REM ENTERING A VALUE OF -9999.99 ENDS THE PROGRAM
40 CLS
50 PRINT@ 256, "DEGREES FAHRENHEIT ";
60 INPUT F
70 IF F = -9999.99 THEN END
80 CLS
90 C = (F - 32) * 5/9
100 PRINT@ 384, F;  " DEGREES FAHRENHEIT = ";  C; "  DEGREES
CELSIUS"
110 GOTO 50
```

Basic Version



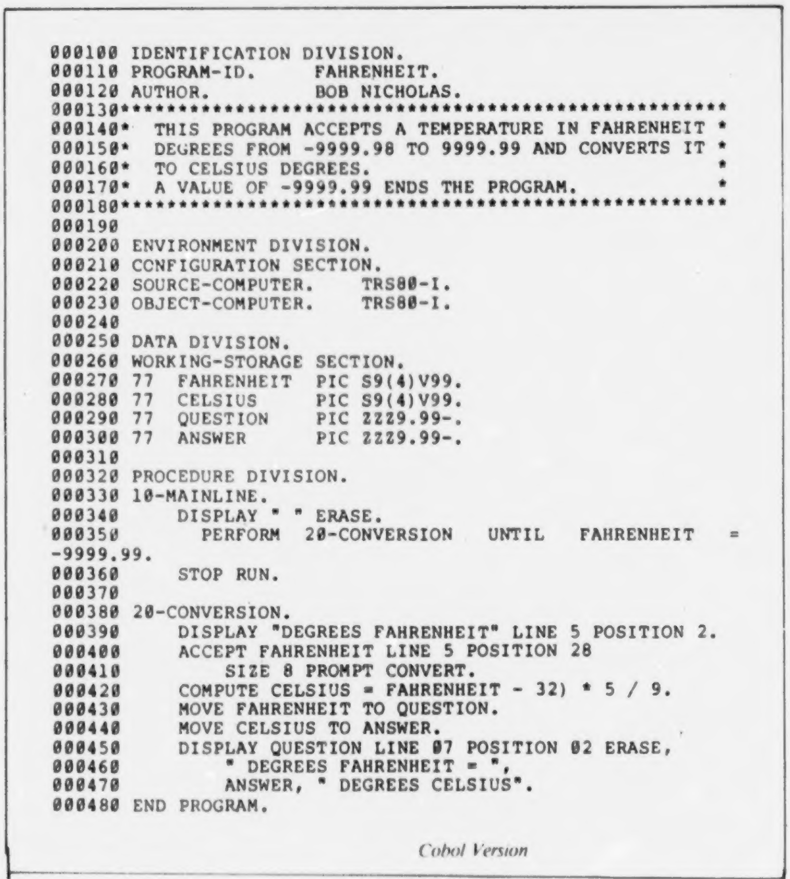
```
00100  C  PROGRAM TO CONVERT FAHRENHEIT TO CELSIUS
00200  C  BY PHIL MARTEL
00300  10       WRITE(1,15)
00400  15       FORMAT(/' TYPE IN THE FAHRENHEIT VALUE'/)
00500           READ(1,20)F
00600           IF (F .LT. -500.) GOTO 200
00700  20       FORMAT(F15.0)
00800           C=5./9.*(F-32.)
00900           WRITE(1,30)F,C
01000  30       FORMAT(1X,G10.3,'  DEGREES FAHRENHEIT IS '
01100         1 ,G10.3,' DEGREES CELSIUS')
01200           GOTO 10
01300  200      STOP DONE
01400           END
```

Fortran Version



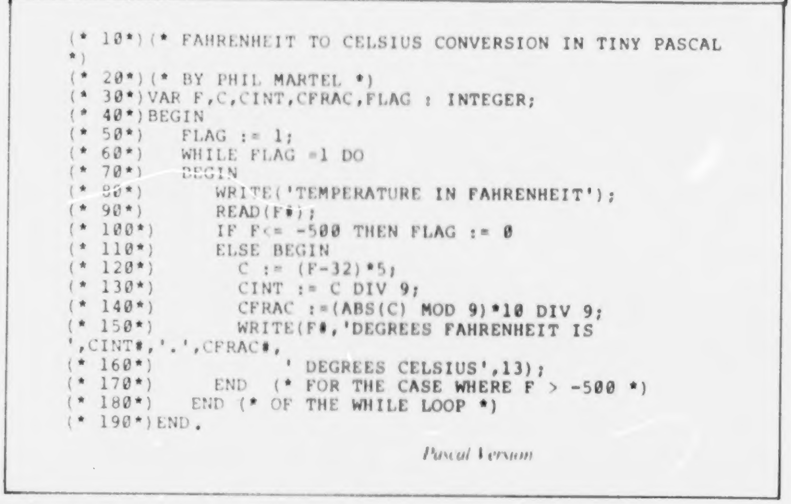
```
0:     CLS
1:     # 28 31


0:     FTOC
1:     CLS
2:     BEGIN  :'ENTER DEGREES FAHRENHEIT ';
3:     F←q
4:     CLS
5:     C←(F - 32) x 5 ÷ 9
6:     F ;'DEGREES FAHRENHEIT = ';C ;'DEGREES CELSIUS'
7:     →BEGIN
```

APL Version



```
000100 IDENTIFICATION DIVISION.
000110 PROGRAM-ID.     FAHRENHEIT.
000120 AUTHOR.         BOB NICHOLAS.
000130*****************************************************
000140*  THIS PROGRAM ACCEPTS A TEMPERATURE IN FAHRENHEIT *
000150*  DEGREES FROM -9999.98 TO 9999.99 AND CONVERTS IT *
000160*  TO CELSIUS DEGREES.                              *
000170*  A VALUE OF -9999.99 ENDS THE PROGRAM.            *
000180*****************************************************
000190
000200 ENVIRONMENT DIVISION.
000210 CONFIGURATION SECTION.
000220 SOURCE-COMPUTER.    TRS80-I.
000230 OBJECT-COMPUTER.    TRS80-I.
000240
000250 DATA DIVISION.
000260 WORKING-STORAGE SECTION.
000270 77  FAHRENHEIT  PIC S9(4)V99.
000280 77  CELSIUS     PIC S9(4)V99.
000290 77  QUESTION    PIC ZZZ9.99-.
000300 77  ANSWER      PIC ZZZ9.99-.
000310
000320 PROCEDURE DIVISION.
000330 10-MAINLINE.
000340     DISPLAY " " ERASE.
000350       PERFORM  20-CONVERSION   UNTIL   FAHRENHEIT   =
-9999.99.
000360     STOP RUN.
000370
000380 20-CONVERSION.
000390     DISPLAY "DEGREES FAHRENHEIT" LINE 5 POSITION 2.
000400     ACCEPT FAHRENHEIT LINE 5 POSITION 28
000410         SIZE 8 PROMPT CONVERT.
000420     COMPUTE CELSIUS = FAHRENHEIT - 32) * 5 / 9.
000430     MOVE FAHRENHEIT TO QUESTION.
000440     MOVE CELSIUS TO ANSWER.
000450     DISPLAY QUESTION LINE 07 POSITION 02 ERASE,
000460         " DEGREES FAHRENHEIT = ",
000470         ANSWER, " DEGREES CELSIUS".
000480 END PROGRAM.
```

Cobol Version



```
(* 10*)(* FAHRENHEIT TO CELSIUS CONVERSION IN TINY PASCAL
*)
(* 20*)(* BY PHIL MARTEL *)
(* 30*)VAR F,C,CINT,CFRAC,FLAG : INTEGER;
(* 40*)BEGIN
(* 50*)    FLAG := 1;
(* 60*)    WHILE FLAG =1 DO
(* 70*)    BEGIN
(* 80*)       WRITE('TEMPERATURE IN FAHRENHEIT');
(* 90*)       READ(F#);
(* 100*)      IF F<= -500 THEN FLAG := 0
(* 110*)      ELSE BEGIN
(* 120*)        C := (F-32)*5;
(* 130*)        CINT := C DIV 9;
(* 140*)        CFRAC :=(ABS(C) MOD 9)*10 DIV 9;
(* 150*)        WRITE(F#,'DEGREES FAHRENHEIT IS
',CINT#,'.',CFRAC#,
(* 160*)                ' DEGREES CELSIUS',13);
(* 170*)      END  (* FOR THE CASE WHERE F > -500 *)
(* 180*)    END (* OF THE WHILE LOOP *)
(* 190*)END.
```

Pascal Version

From Nicholas, Robert, and Philip Martel, "TRS-80 Tongues," <u>80 Micro,</u> December, 1982, pp. 216-24.

EXERCISE (ASSIGNMENT):

- Select a computer language mentioned in class. Look up information about this language and write a paragraph that explains its history or use.

Lesson 36

COMPUTER LANGUAGES/BINARY NUMBERS

PERFORMANCE OBJECTIVES: Students will be able to:

- recognize the relationship between binary numbers and internal data storage.
- distinguish between machine language and higher-level languages.
- name at least four higher-level computer languages.

VOCABULARY: NEW: acronym, bit, byte, kilo (k), "word," notation, BASIC, FORTRAN, COBOL, Pascal, compiler.

MOTIVATION:

- Using the word PRECAUTION coded from 0-9 as shown, find the word equivalent to each numerical value.

P	R	E	C	A	U	T	I	O	N
0	1	2	3	4	5	6	7	8	9

  - 3416
  - 146
  - 0881
  - 986732
  - 61432

- What does this word game have to do with computers? (A similar process takes place to interpret and store information in a computer. We start with the BASIC language which must be transformed into a numerical code that the computer recognizes and understands.)

AIM: How do we communicate with the computer?

DEVELOPMENT:

1. What two commands do a flashlight or a television understand? (On and off.)

2. How can we assign number values to on and off? (Zero is off, one is on.)

3. What number system is this? (Binary number system or base two, since it is based on the two digits, zero and one)

4. Since in binary, we can't write two as 2, how is the number two represented? (10)

5. Let's write a chart for the numbers zero to nine:

Arabic Numerals	Binary Code
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001

- Let's see how positioning works in binary:

	sixteen	eight	four	two	one
21 =	1	0	1	0	1

- How do we move from one to two to four to sixteen, etc.?
  (Starting from base 2, or $2^0 = 1$, we move to 2^1, 2^2, etc.)

- $$\frac{2^7}{128} \quad \frac{2^6}{64} \quad \frac{2^5}{32} \quad \frac{2^4}{16} \quad \frac{2^3}{8} \quad \frac{2^2}{4} \quad \frac{2^1}{2} \quad \frac{2^0}{1}$$

- How do we figure out which number in binary is what number in our usual decimal representation? We add the value of all columns containing one and ignore all zeros.)

- Examples:
  - 17 = 10001
  - 5 = 101
  - 34 = 100010

6. How does this relate to on and off? (0 and 1 represent OFF and ON states (electrical circuits, open/closed gates, or magnetized/demagnetized states.)

7. What computer terminology is derived from this binary system?
   - bit - binary digit
   - byte - usually 8 bits ("unit" of information)
   - K - 1000 (kilo) - really $2^{10} = 1024$
   - 32K Bytes - 32 x 1024 bytes of data (Memory space)
   - Word - 2 bytes

8. Why don't we write programming statements in binary notation if that is what the computer understands? (This would be too complicated for humans, so we use higher-level languages which are converted to binary inside the machine by something called a compiler.)

9. What are some higher-level languages?
   - **BASIC:** Beginners All Purpose Symbolic Instruction Code (1963)
   - **FORTRAN:** Formula Translation (1955)
   - **COBOL:** Common Business Oriented Language (1956)
   - **PL/1:** Programming Language 1
   - **APL:** A Programming Language
   - **Pascal:** named after Blaise Pascal
   - **Ada:** used by Department of Defense and named after Countess Ada Lovelace

   - NOTE: Most languages are known by acronym, a word formed from the first or first few letters of several words.

APPLICATION:

- If A = 1000001, write a chart of the alphabet and the binary notation for each letter.

- Write the binary notation for the following numbers:
  - 63
  - 45
  - 100
- Write your age in binary notation.

SUMMARY: Why have languages such as BASIC been developed?

EXERCISES (ASSIGNMENT):

- What decimal number has the binary representation 10101?
- Using the alphabet chart we devised, write the following BASIC commands in binary.
  - LET
  - GOTO
  - LIST
  - RUN

Lesson 37

CONSUMERISM — PURCHASING A PERSONAL COMPUTER

PERFORMANCE OBJECTIVES: Students will be able to:

- identify the components available to purchasers of a personal computer system.
- perform a needs-assessment to determine the purpose for which the computer might be used.
- explain the concept of upgrading from a minimal configuration to one more advanced.
- balance the various factors involved in selecting a personal computer and arrive at an appropriate decision.
- list the methods of evaluating a personal computer system.

VOCABULARY: NEW: needs assessment, upgrade, evaluate

REVIEW: keyboard, monochrome monitor, color monitor, processor, printer, disk drive, memory capacity, modem, graphics

MOTIVATION:

- Distribute Worksheet. Adequate copies must be reproduced beforehand.
  - Allow adequate time for students to familiarize themselves with the document.
  - If you were a salesperson, which system would you recommend for each of these three customers?
  - Why would System B be wrong for Margaret?
    Why would System C be wrong for Julio?
    (Those would not meet their needs.)
- What lesson can be drawn from this exercise? (Purchaser must choose carefully; purchase must fit needs.)
- We just learned something about buying sound equipment. However, we have been studying computers! What connection is there? (Same rulles apply to purchasing computers. Elicit Aim.)

AIM: How can we learn to be intelligent comsumers when purchasing a personal computer?

DEVELOPMENT:

1. This lesson will help us to develop some guidelines for the time when we are ready to purchase a personal computer.

2. We just learned that the purchaser of a sound system must be careful to match the system to his or her purposes or needs.

   . There are many purposes for which a person might buy a computer.

   . How might a computer provide entertainment? (Various computer games are available.)

   . How might a computer help us to manage the house? (balance checkbook, keep financial records, keep file of recipes, addresses, etc.)

   . How might a computer help us to learn? (programs to teach skills: language, typing, etc.)

   . How might a computer help us with our writing? (word processing)

   . How might a computer allow us to be creative? (creating programs, graphic displays, etc.)

   . How might a computer provide information? (connect to electronic information systems)

3. Since a computer can do so many different things, what is the first question that a purchaser should ask? (How do I plan to use this computer? How might I use it in the future?)

4. Here is a list of computer components that you could purchase. (Place on board.)

   Keyboard, monochrome display, color monitor,
   processing unit, disk drive, printer, tape drive
   telphone hook-up (modem), expanded memory capacity

   . Which of the above components would a purchaser need if the computer were to be used for:

     . playing games
     . learning to play games
     . keeping personal records
     . word processing
     . learning new skills
     . using an electronic information service

   . Allow students to suggest various combinations of equipment. Emphasize the concept of multiple solutions.

5. Suppose that you purchased a computer just to play games. However, a year later you realized that you wanted to use the computer both for word processing and to keep personal files of information. How might you solve this dilemma? Purchase the additional equipment needed. This is called upgrading.)

   . What might happen if you were to discover that this computer could not operate additional equipment? (You would be at a disadvantage.)

   . What might a smart purchaser do in order to avoid unpleasant surprises in the future?
   (Be sure that the computer can be upgraded.)

   . How can a purchaser find out if the computer can be upgraded in the future?
   (Ask salesperson, talk to people, read.)

6. Suppose you had "X" dollars to spend on a computer system. There are three systems in which you are interested, as follows: (Place on board)

   System-X: Relatively inexpensive model. It has keyboard, processor, and monochrome display only. The company promises disk drive, printer, and other expansion devices in the future. It seems more like a toy then a computer.

   System-Y: Well-known brand, but expensive. Your dollars will buy only keyboard, processor, and monochrome display. Many other devices available now. Computer has excellent reputation.

   System-Z: Your dollars will buy keyboard, processor, color display, printer, and disk drive. Unfortunately, manufacturer has only recently entered the computer business.

   . Allow for open but orderly debate. Emphasize the possibility of multiply-correct solutions. Discuss the concept of balancing factors to arrive at a decision.

7. Making a decision such as the one above is difficult. Sometimes people make a decision and then regret it. Is there any way to find help in making such a decision? (Talk to friends who are knowledgeable.)

   . To whom else might you turn for help? (Salesperson in dealership.)

   . Where might you find a totally unbiased review of various computers?
   (Read computer magazines and consumer-oriented magazines.)

   . How might you determine if the computer "feels right"?

   (Ask to use the computer at the store.)

SUMMARY:

What are some rules that we have developed to help us become better-informed and more intelligent computer consumers?

- Decide reason(s) for buying a computer.
- Decide how sophisticated the system should be.
- Find out if the system can be upgraded in the future.
- Decide how much to spend and how to get the most for our money.
- Research systems by talking and reading.
- Actually try computer to determine suitability.

EXERCISE (ASSIGNMENTS):

- Distribute Worksheet - II (Adequate copies must be reproduced.)
- This is a long-term assignment rather than an overnight assignment. Establish a reasonable deadline for findings to be presented. Explain that findings can be obtained from a variety of sources, such as computer magazines, newspaper advertisements, and personal visits to computer dealerships. Encourage students to work in committee. Anticipate and answer questions concerning terms and format of the worksheet.
- (Optional) Interview a person who has purchased a personal computer. Obtain answers to the following:
  - How will the computer be used (purposes)?
  - Which of the various components were purchased?
  - How will the system be upgraded in the future?
  - How did the person decide to purchase this particular brand?
  - How much research did the person do before making the purchase?
  - How satisfied is person with the purchase? What would the person do differently if she could re-make the purchase?
  - Would the person recommend this particular brand to the others? Why or why not?
- Provide a list of names and addresses of computer manufacturers. Have student select one and write to the manufacturer for information on this particular brand of computer, using correct form for a business letter. Request information concerning technical details and current prices.

WORKSHEET - I

Instructions: Read this worksheet carefully and arrive at the decision requested. You and your classmates will discuss your decisions as part of the lesson.

Three customers have come into Weird Melvin, a famous stereo and audio dealer. Each customer intends to purchase a sound system that would suit his or her needs. Can you help them by matching the systems described in Column B with the customers described in column A?

Column A

Julio - a construction worker. He wants to be able to listen to his favorite music while he works.

Margaret - a professional musician. She was trained at the Juilliard School of Music. She requires absolute perfection and realism. Money is no object.

Donald - a college freshman. He wants a system that he can use both in his dormitory room and also in his room at home. He wants a fairly realistic sound. He has a moderate amount of money to spend.

Column B

System "A" -
- automatic record changer
- 11-inch bookshelf speakers
- AM/FM mini-tuner and amplifier
- mini-stero cassette player which connects to amplifier

System "B" -
- one-piece AM/FM stereo cassette player with attached earphones.

System "C" -
- twin speakers, each 35 inches high
- dual stereo tape deck with push-button controls and LED meters
- linear-tracking direct-drive turntable
- digital synthesized, 100 watt per channel receiver
- 7-band equalizer with LED read-out

WORKSHEET - II

Instructions: This worksheet will be used to evaluate two different brands of personal computers. You may obtain your answers by reading newspaper advertisements or magazine advertisements and by visiting stores in which the computers are sold.

	Answers:	
Information to be found	Brand - 1 Name: ______	Brand - 2 Name: ______
1. Cost of Basic Computer (keyboard, processor)		
Monitor included?		
If extra, cost of monitor		
2. Memory Size (in "K" or Kilobytes)		
RAM (usable memory)		
ROM (non-usable memory)		
Can memory be upgraded?		
Cost to upgrade		
3. Color Capability (yes/no)		
How many colors available?		
How many colors displayed at one time?		
4. Graphics Capability (yes/no)		
High-resolution graphics?		
Special graphics commands?		
5. Peripheral Devices		
Disk Drive included?		
Disk Drive available?		
Cost to add Disk Drive		
Cost of Tape Drive		
Printer Available?		
Cost		
6. Software		
Type of Software available		
Average cost of software		

Lesson 38

WORD PROCESSING

PERFORMANCE OBJECTIVES: Students will be able to:

- gain a working knowledge of a word processing program.
- understand the value of a word processing program.
- understand the impact of the word processing program on the world around them.

VOCABULARY: save, load, delete, insert, edit, cursor, files, revise, store, retrieve, text

MATERIALS:

Word processing program, revised documentation, poster (optional), chart of editing commands for the word processing program being used, copies of editing commands for the program being used.

MOTIVATION:

. What is word processing?
. Distribute attached worksheet. Read selection.
. What is a word processing program?
. What does it do?
. What is it used for?
. Who uses these programs?
. What can the program do for me?
. How can I use this program now?
. How will this kind of program be used in the future?

AIM: How do we use a word processing program?

DEVELOPMENT:

1. Demonstrate how to load word processing program into the computer.

2. Have students write several sentences.

3. Explain basic operation of editing procedures.

4. Allow students to practice editing the text on the screen.

5. Students assemble in groups to create a continuing story. One student begins by typing in a sentence. The second student adds his sentence. Continue process until all students have typed sentences.

6. Students decide how to improve the story they have created. They should edit the text and print out the completed copy for display.

APPLICATIONS:

- Write a dialogue between two classmates.
- Write a poem.
- Create activity cards:
    - Story starters for creative writing.
    - Write a new ending to a familiar story.
    - Write a dramatic dialogue.
    - Describe a favorite character from a book or television.

SUMMARY:

- What are the advantages of word processing programs?
- How do we use word processing programs?

EXERCISES (ASSIGNMENTS):

- Create a word processing history book.

  Students can research the history of the word processor and trace it to the present time. Information can be obtained from computer manufacturers or software companies.

- Research:

  Where are word processors being used today? Students can contact large companies (insurance, brokerage, banks, etc.) for information about what position or jobs require word processing knowledge. This information can be consolidated and discussed.

- Edit poem or passage written in class. Print out a copy for a class magazine.

Worksheet

What is a word processing program?

A word processor is a computer program that makes it easier to type, revise, store, retrieve, and print text. When you use a word processing program on your computer and you are typing at the keyboard, whatever you type appears on the screen and not on a piece of paper. At the same time as your writing appears on the screen, it is being stored in the computer's memory. (This area is called the "workspace".) Because your text is being stored in the computer's memory, you can move forward or backward through the text to read what you have written and to edit and revise the material. You can use the editing functions of the program to make changes in the text while it is in the memory.

In order for you to be able to use your material again, you must save it on disk or cassette. Remember, when you turn the machine off you erase what is in the computer's memory (or workspace).

Your material can be reloaded at any time into the memory of the computer so you can continue to work on it or have your word processing program print out a copy for you. Remember, you must first load your word processing program into the computer before you can continue to work on your material.

Lesson 39

DATA BASE MANAGEMENT

PERFORMANCE OBJECTIVES: Students will be able to:

- gain a working knowledge of a data base management program.
- explain the value of a data base management program.
- discuss the impact of a data base management program on the world around them.

VOCABULARY: NEW: file, record, data base management

MATERIALS:

Data Base Management program, documentation

MOTIVATION:

- How do your teachers know what mark to give you at the end of the term? (They keep records of all test and other items in a marking book and use them to decide on a mark).
- How could a computer be used for the same purpose?

AIM: What is a Data Base Management program?

DEVELOPMENT:

1. Demonstrate how to load a Data Base Management program into the computer.
2. The first thing we have to do is create the file so that each record has the information we need.
   - What information would we need for a marking book? (Student's name, class, test 1, test 2, homework, report)
3. Set up file with students so that each record will contain the categories needed.
4. Access record #1. What appears? (A blank record appears. The categories listed when we created the file are displayed, and there is room to enter the information for record #1.)
5. Type in the entries; verify that the information is correct; and close the file.

6. Go on to record #2. Have a student enter the data for himself.

7. Continue as above. Have students take turns entering data.

8. Demonstrate how to change information in a record. Let a student raise a test grade on his/her record.

9. Demonstrate how to print out an individual record.

10. Demonstrate how to print out with specifications (how the printout should look). Use print options.

11. Aside from marking books for teachers, what other uses are there for Data Base Management programs?

    - Elicit suggested applications from class and place list on chalkboard.

    - Examples:

      Travel Agency - airline, bus, and train schedules

      Library - catalog of books, magazines, newspapers

      Supermarket - inventory of goods, list of employees

      Police Department - list of criminals, robberies

APPLICATIONS:

- Create a file for a travel agent so that a person could find a flight from Chicago to Boston on the morning of April 14. Use the information in the chart below.

Airline	Flight No.	Leave	Arrive	Class First Coach(Y) Single(S)	Meal	No. of Stops	Seats Available First	 Coach(Y)
American	332	9:30a	3:06p	F/Y	Snack	3	6	3
United	252	10a	1:46p	S	Lunch	1	11	
Allegheny	197	10:15a	2:42p	F/Y	Lunch	2	7	9
TWA	548	11a	1:58p	F/Y	Snack	0	2	4
American	280	11:20a	4:36p	F/Y	Lunch	3	7	8
Allegheny	212	11:35a	2:32p	S	Snack	0	4	5

Fare:	One Way	Round Trip
First Class	74.00	148.00
Single Class	59.00	118.00
Coach	59.00	118.00

- Use the file to arrange trips for class members.
- Expand the file to include other flights. Set up the records so that you can get information on any item you wish.

SUMMARY:

- What are the advantages of Data Base Management programs?
- How will they be used in the future?

EXERCISES (ASSIGNMENT):

- Create an inventory file for your local grocery store.
- Set up a record keeping system for any purpose that you determine.

Lesson 40

ELECTRONIC SPREADSHEET

PERFORMANCE OBJECTIVES: Students will be able to:

- gain a working knowledge of an electronic spreadsheet program.
- explain the value of an electronic spreadsheet.
- discuss the impact of electronic spreadsheets on the world around them.

VOCABULARY: NEW: spreadsheet, financial planning, budgets

MATERIALS:

electronic spreadsheet program, documentation.

MOTIVATION:

- We are going to open a class snack store. What do we have to do to figure out our expenses and profits?
- Elicit items such as rent, phone, labor, advertising, equipment, cost of purchasing items, charge to customers, etc.
  - If we consider some of these items expenses and the others income, how do we keep records?
    - Add up expenses.
    - Add up income.
    - Subtract expenses from income to find out what the profit is.
  - What advantages would there be to using a computer for this purpose?

AIM: What is an electronic spreadsheet program?

DEVELOPMENT:

1. Demonstrate how to load a electronic spreadsheet program into the computer.
2. What appears on the screen? (A blank form with column and row headings appears.)
3. Let's use the rows to list all of the items that are expenses. Include a listing for the total.

4. Let's enter the amounts for the first month in the first row.

5. How can we use the computer's ability to perform as a calculator to get our total of expenses? (Demonstrate the formula function of the spreadsheet.)

6. Now let's create a list of income items, enter the amounts, and use a formula to get the total.

7. How do we know what our profit is? (Create a listing and formula for profit.)

8. How well is our snack shop doing?

9. If we raise the price of potato chips and soda, how do we show that on the computer?

   Note: Demonstrate how to enter item changes and point out the totals are affected by the changes.

10. How can we use the other rows?
    Note: Demonstrate weekly or monthly breakdowns.

11. What uses are there for electronic spreadsheet programs?

    - Elicit suggested applications from class and place list on chalkboard.

    - Examples:

    Bank - check processing, savings account recordkeeping, investments.

    Hospital - patient billing, operating room schedules, patient medication records.

    Store - inventory control, customer billing, sales commission.

APPLICATIONS:

- Create a payroll report for a local business. Use the information in the chart below in your spreadsheet.

Payroll for Week Ending 04/27/84

					Deductions						
Employee	Hours Worked	Hourly Rate	O/T	Gross Pay	FICA	Fed Tax	State Tax	City Tax	Disab Ins.	Total Ded.	Net Pay
Gaines	41	7.25	1	300.88	19.56	30.09	9.03	4.52	.30	63.50	237.38
Henry	39	7.50	0	293.00	19.05	29.30	8.79	4.40	.30	61.84	231.16
Ingram	44	6.85	4	301.40	19.59	30.14	9.04	4.52	.30	63.59	237.81
Jones	36	8.10	0	291.60	18.95	29.16	8.75	4.38	.30	61.54	230.06
King	40	7.85	0	314.00	20.41	31.40	9.42	4.71	.30	66.24	247.76

- Add class members to this list and include their information to this payroll.
- Give someone a raise and note how the other items for this person change.

SUMMARY:

- What are the advantages of electronic spreadsheet programs?
- How will they be used in the future?

EXERCISE (ASSIGNMENTS):

- Use the information given in the class application to generate monthly and yearly payroll reports for the company.
- Create an inventory control for the company. Keep a record of all items the company purchases and sells.
- For each item that the company purchases and sells, create an accounts payable and accounts receivable statement.

Appendices

Appendix A

Individual Student Projects

1. Employees of the Tiger Electric Company are given raises as follows: Salespeople: 12%; Production Workers: 8%; Managerial Staff: 15%. Given an employee's job category number (1, 2, or 3), print the employee's new weekly salary.

   Input: Job category number, old salary, name
   Output: Job category number, new salary, name

   Run this program for 5 different employees.

2. Write a program to calculate a person's gross weekly pay, given the rate per hour, overtime rate per hour, number of regular hours worked that week, and number of overtime hours that week.

   Run this program for 5 workers.

3. Assume that one ounce of hamburger contains 80 calories, a bun has 200 calories, and one french fry has 14 calories. Given the number of ounces of hamburger, whether a person has a bun, and the number of french fries, calculate the number of calories in a person's meal. Generalize this program for any group of foods.

   Run for 5 different meals.

4. If a student's final average is:

95 or over	Principal's List
90 - 94	Honor Roll
85 - 89	Merit Roll

   Write a program to read students' grades in 5 subjects, average them, and print out which categories they belong to.

   Run this for 5 students.

5. Given the age of a person and the day of the week his birthday falls on this year, write a program to print the day of the week he was born. (Watch out for leap years.)

6. Produce a table listing the Fahrenheit temperature for each Celsius temperature from 0° to 100° in increments of 5° C. Use the formula $F = \frac{9C + 32}{5}$

   For what temperature does Fahrenheit = Celsius?

7. Devise a program to decode messages written in a code you create. To test the program, teach a friend your encoding scheme and have him or her encode a message without showing it to you. The computer should decode this message correctly.

8. Which would you rather have $1,000,000 or one cent a day doubled for 30 days (i.e., .01 + .02 + .04 + ...(thirty times)? Find the difference in these two amounts.

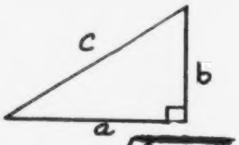
9. Write a program to play the following game. The computer tries to guess a number you have in mind from 1 to 100. First, it guesses a number and you tell it if the number is too high, too low, or correct. On the basis of the information you give, the computer guesses again. This continues until the computer guesses right.

10. Write a program to print a tic tac toe diagram on the screen. Use as few steps on the program as possible.

Project (INPUT Statement and Arithmetic Operations)

1. In this program, you will input two sides of a triangle (legs), and the computer will print the length of the hypotenuse. The triangle, of course, is a right triangle.

2. Use the PYTHAGOREAN THEOREM: In a right triangle, the square of the hypotenuse's length equals the sum of the squares of the lengths of the legs.



3. Since $c^2 = a^2 + b^2$, $c = \sqrt{a^2 + b^2}$

4. Screen printout is indicated.

```
SIDE 1        3
SIDE 2        4
HYPOTENUSE    IS  5

SIDE 1        5
SIDE 2        12
HYPOTENUSE    IS  13
```

User Types

Computer types everything else

etc.

5. Use the following data values for legs.

SIDE 1	SIDE 2	
6	8	
9	12	
10	24	
15	36	
20	21	
5E1	5E1	(Isoseles)

6. Have the computer skip a line between each example.

7. Using the computer in the calculator mode, check the accuracy of your results.

Due Date:

Penalty: 5 points/week for each week late.

Project (FOR/NEXT Statement)

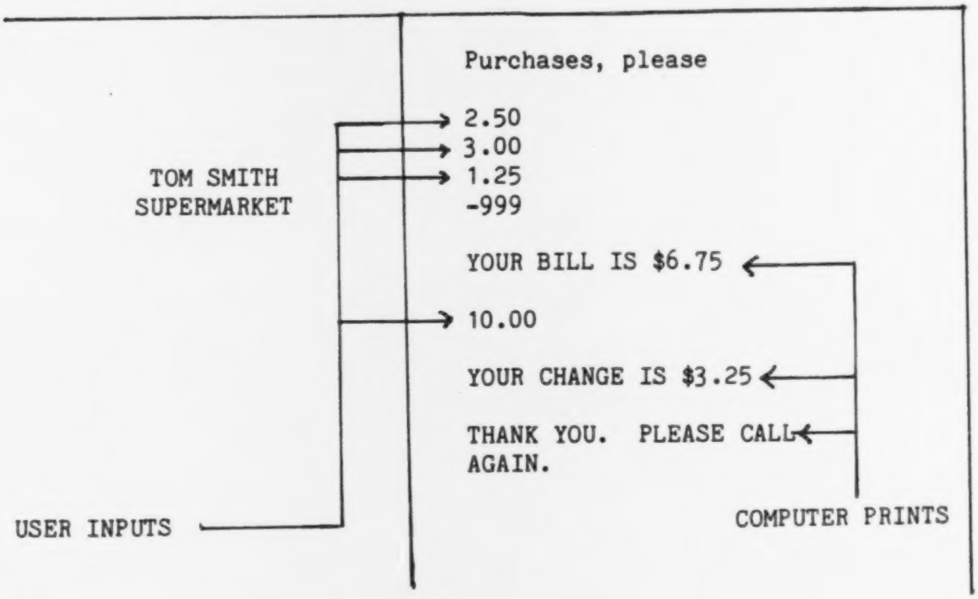
1. Screen clears. The name of your "SUPERMARKET" appears on the screen.
2. Use a FOR-NEXT loop counting to 2,000 as a time delay -- make this a subroutine, if you wish.
3. Screen clears again.
4. Computer prints: "Purchases, please"
5. You input various amounts, one per line. As you input them, the computer adds them.
6. When you input a sentinel data value of -999, the computer prints the total as follows:

   YOUR BILL IS $_______.___

7. You then input the amount of money the customer gives you.
8. The computer calculates the change owed the customer and prints:

   YOUR CHANGE IS $_______.___
   THANK YOU. PLEASE CALL AGAIN.

9. The program repeats from step 3.



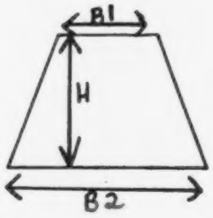
```
Purchases, please

2.50
3.00
1.25
-999

YOUR BILL IS $6.75

10.00

YOUR CHANGE IS $3.25

THANK YOU.  PLEASE CALL
AGAIN.
```

Project (INPUT Statement)

1. The area of a trapezoid is given by the formula:

   $$A = \frac{1}{2}H(B1+B2)$$

   where H is its height, B1 is top base and B2 its lower base.



2. Write a program where you input values for H, B1, and B2. Calculate the area of the trapezoid and print out its value.
3. Trace the program manually to make sure it's correct.
4. Use the INPUT statement to print out messages, so that the computer tells the user what to input on each line. The output should look like this:

```
              HEIGHT ⑥                    User INPUTS
Computer      TOP BASE ④                  these values
types         LOWER BASE ⑧                (see table
              AREA OF TRAPEZOID IS ㊱       below)

Next          HEIGHT  4                   Computer's
Set of        TOP BASE 7                  answer
Data
```

5. Skip a line between examples.
6. The program repeats.
7. Type in the listed values for H, B1 and B2.

H	B1	B2	Calculated Area	Computer's Answer
6	4	8		
4	7	11		
16	25	75		
3	10	30		
10	30	50		
12.3	5.5	3.2		

Due Date: ____________________

(Penalty: 5 points/week for each week late)

Appendix B

Supplemental Strategies for Reinforcement

Supplemental strategies for teaching:

- PRINT Statement using Strings
- PRINT Statement using Numbers
- Using Mathematical Operators
- Order of Operations
- Use of Parentheses in Operations
- Use of the Comma in a Print Statement
- Use of the LET Statement for String Variables
- Use of the LET Statement for Numeric Variables
- Use of the INPUT Statement for String Variables
- Unconditional Branching
- Counting
- READ/DATA
- IF/THEN

ADDITIONAL ACTIVITIES FOR PRINT STATEMENT

1. Direct the students to type in:

   HI COMPUTER and press the ENTER (or RETURN)

2. Ask, "What was the result?" (Syntax error)

3. Tell the students that a syntax error is the computer's way of telling you that it did not understand what you just typed.

4. Explain that because computers do not have the capability to understand English, we therefore have to use their language. The part of their language that tells the computer to print what you want is the word PRINT with the use of quotation marks, "....." (Show "   " on the keyboard)

5. Try this:
   PRINT "Hello, Computer"

6. Ask, "What happened?"

CHALLENGE:

1. Make the computer display your name, address, city, state, and zip code on the screen.

PRINT STATEMENT:

1. Review print "string" concept. Emphasize need for "string" for message.

2. Give examples such as:
   PRINT "   "

3. Introduce PRINT "1 + 3"
   Discuss the output.

4. Introduce PRINT 1 + 3
   Discuss output. Ask why the computer calculated the answer rather than printing 1 + 3 message.

5. Review PRINT "________" and
   PRINT 1 + 5 (substitute numerical values)

6. Have class do the following and compare results

   a. PRINT "5 + 3"
      PRINT 5 + 3

   b. PRINT "10 + 10"
      PRINT 10 + 10

   c. PRINT 25 + 10
      PRINT "25 + 10"

7. Review the differences between results. Trace the output of each statement.

8. Ask class to determine the output of each:

   a. PRINT "Hello"
   b. PRINT "1 + 1"
   c. PRINT "My name is John"
   d. PRINT 5 + 5
   e. PRINT 10 + 5
   f. PRINT 6 + 4

   After these above concepts are mastered, introduce:

9. Type this: PRINT My name is Sue.

10. Discuss the output.

11. Explain that without "   " the computer is expecting to see a number, otherwise the computer will print a zero.

12. Try others, i.e., PRINT "Hi, There"
    Discuss results. PRINT, Hi There.

USE OF MATHEMATICAL OPERATORS

1. Review the use of the PRINT statement in addition. Discuss how the computer acts as a calculator.

2. Discuss (put on board or chart)

+	Tells the computer to add
-	Tells the computer to subtract
*	Tells the computer to multiply
/	Tells the computer to divide

   Show keyboard. Demonstrate where to locate +, -, *, /.

3. Demonstrate on the computer:

```
PRINT 7+5
PRINT 7-5
PRINT 7*5
PRINT 7/5
```

4. Remind the class that the computer does not need "=" sign in order to calculate.

5. Allow the students to determine the following answers without the use of the computer.

```
PRINT 10 + 2
PRINT 12 + 7
PRINT 12 - 3
PRINT 6 - 5
PRINT 4 * 3
PRINT 5 * 4
PRINT 10/2
PRINT 50/5
```

   Then check their answers on the computer.

6. Review keyboard location for +, -, *, /.

7. Review rules to make computer act as a calculator.

CHALLENGE:

1. Have the computer calculate your average test score. You received these grades: 85, 90, 82, 76, 93 and 89.

ORDER OF OPERATIONS

1. Review use of operators and their location on the keyboard.

2. Discuss In BASIC, the computer evaluates an expression according to the priority rules.

   - First multiply and divide from left to right.
   - Then add and subtract from left to right.

   For example:

```
PRINT 2 + 3 - 1
          5 - 1
              4

PRINT 3 + 7 + 5
         10 + 5
             15

PRINT 2 * 5 + 3
         10 + 3
             13
```

   Now try these on the computer.

3. Determine the answers to these without the computer. Remember the Left-Right Rule.

```
PRINT 8 * 2 -  1
PRINT 10/5 + 6
PRINT 20/10 + 8
PRINT 16 - 5 + 10
PRINT 4/4 + 1
PRINT 10 * 5/25
```

4. When the Left-Right concept is understood, introduce the concept that * and / will be calculated first (again in a Left-Right order).

```
PRINT 10 + 6/3
```

- Discuss the answer. Illustrate how the computer did the calculation

```
PRINT 10 + 6/3
      10 + 2
          12

PRINT 3 + 5 * 3
      3      15
             18
```

- Keeping in mind the left to right rule, have students determine the answer to these:

```
PRINT 5 + 4 * 6
PRINT 2 * 3 * 3
PRINT 8 * 2/4
PRINT 20/5 + 6
```

- Discuss how the calculations were made. Show step by step calculations.

5. When the * and / first rule is understood, introduce the next order of operation + and -.

   Keeping in mind the Left-Right rule, show how these are calculated:

```
PRINT 5 + 10 * 2
      5 + 20
          25

PRINT 6 -  5 * 4
      6 - 20
        - 14
```

6. Review

   a. Left to Right rule

   b. * and / are calculated first from Left-Right then

   c. + and - are calculated next from Left-Right.

7. Ask the class to trace the order of calculating

```
PRINT 1 + 2 * 3 + 6 - 4
      1 +   6   + 6 - 4

      1 + 6 + 6 - 4
        7   + 6 - 4

      7 + 6 - 4
        13  - 4

      13 - 4
        9
```

8. Give other examples to class. Have them trace the order.

```
PRINT 3 + 5/5 + 6
PRINT 9 - 10 * 4 + 11 + 3
```

USE OF COMMA TO PRINT OUTPUT IN COLUMNS.

1. Review the PRINT "String" and PRINT "numeric" statements.

2. Discuss when we have more than one item in a print statement such as:

   - PRINT "A", "B", "C", "D"

   - We use commas to stretch apart the output. Have class type in:

     PRINT "A B C D" and
     PRINT "A", "B", "C", "D"

   - Discuss the output. Review that commas stretch out the answers and puts them in columns about 16 spaces apart.

3. Discuss the use of columns. Demonstrate, having the class type:

   - PRINT "A", "B", "C", "D", "E", "F", "G", "H", "I", "J"

4. Review: To get the computer to put our answer in columns, we need commas outside the quotation marks or between the parts of a string.

5. Try these:

   - PRINT "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"
   - PRINT 1 + 1, 2 + 2, 3 + 3, 4 + 4, 5 + 5, 6 + 6, 7 + 7, 8 + 8
   - Illustrate the lack of quotes in the second example and how the commas put numerical answers in columns too.

6. Challenge

   - Have the computer display a day in your class schedule in one column and your teachers names in another column.

USE OF PARENTHESES:

1. Review location of operators on the keyboard.

2. Review operators for functions.

3. Review order of operations.

4. Have the class do the following computations the same way the computer would:

```
PRINT 5 * 3 + 2
PRINT 10/5 - 3
PRINT 25 * 3 - 50
PRINT 3 + 10/2
```

   Show the steps involved in each computation.

5. Have the class do the same computations, but include parentheses as shown:

```
PRINT 5 * (3 + 2)
PRINT 10/(5 - 2)
PRINT 25 * (3 - 50)
PRINT (3 + 10)/2
```

6. Compare the results. Ask the class why the computer gave them different answers.

7. Tell the class that when an operator is between parenthesis, the computer will calculate that operation first and then go on to the operators outside the parenthesis.

8. Have the class do the following on paper

```
PRINT 8 + (2 - 5) * 4
PRINT (8  + 2) - 5 * 4
PRINT ((8 + 2) - 5) * 4
PRINT (8 + 2 - 5) * 4
PRINT 8 + 2 - 5 * 4
```

   Explain step by step calculations.

9. Let the students compare their answers with the computer's calculation.

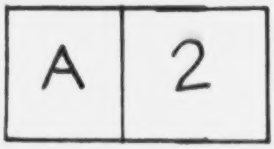
10. Have students place parentheses around different parts of this statement.

    - PRINT 3 + 4 * 5 + 2 - 7 and let the computer calculate different answers.

11. Review the use of parentheses for calculations.

VARIABLES (Numeric)

1. Tell class to imagine that inside the computer are lots of little boxes with two sides. They look like



   The left side is for a variable name. The right side is for the value of that variable.

2. Illustrate

Variable	Value
A	2

- Ask: "What is A's value?"
  "How do you write it?" (A = 2)

3. Ask the same questions for these variables:

C	9
D	3.5
E	0
X	8
Y	-1
Z	15

4. Ask the same question in reverse. "What variable has the value of 9," etc.

5. Review and define <u>variable</u> and <u>value</u>.

6. Explain that in BASIC we can use the statement LET (a variable) = (any value) to give a variable a value.

7. For example, if we want to give variable C a value of 9, we write it as:

   - LET C = 9

8. Elicit the BASIC statement for other variables and values on the chart.

9. Have class write a program that assigns values to the variables and prints the values. Such as:

```
10    LET C = 9
20    LET D = 3.5
30    LET E = 0
40    LET X = 8
50    LET Y = -1
60    LET Z = 15
70    PRINT C, D, E, X, Y, Z
80    END
```

10. Explain that the print statement does not tell the computer to print C, D, E, X, Y, Z because they do not have quotes around them. Instead, the computer prints the values of the variables C, D, E, X, Y, Z.

11. Have the class fill in the value of the variable from the program below:

```
10    Let A = 2
20    Let B = 5
30    PRINT A + B
```

- Explain that the values for A and B do not change even though the calculation was made by the computer.

12. Try:

```
10    LET X = 10
20    LET Y = 4
30    LET Z = X * Y
40    PRINT Z
```

- Explain each step.

VARIABLES (Strings)

1. Review definition of variable and value.

2. Explain that variables can be a number or a string.

3. Review need for "   " when using strings.

4. Explain when we are using strings variables, the variable name must end with "$".

5. Demonstrate location of "$" on keyboard.

6. Explain that the string value is enclosed with quotation marks just as in the PRINT statement.

7. Demonstrate:
Let A$ = "This is easy"

9. Make a chart as follows:

Let B$ = "This is fun"

Let C$ = "I can do it"

LET D$ = "I like this"

10. Ask what is the value of the variable B$? of C$? and of D$?

11. Ask what variable has the value:

    - This is fun?
    - I like this?
    - I can do it?

12. Have the class fill in the values from this program.

```
10    Let X$ = "I"
20    Let Y$ = "Understand"
30    Let Z$ = "Computers"
```

13. Challenge

    - Write a program using the LET statement that prints your name, address, city, state, and zip code.

INPUT

1. Review definition of variable and value.

2. Explain that another way of putting variable names in boxes and giving them values is by using the INPUT statement.

3. Example:

```
10    INPUT A$
20    INPUT B$
30    PRINT A$, B$
```

    - Ask what is the value of A$ and B$. Elicit the fact that it is not known.

4. Therefore, the computer will ask you by printing a "?".

   - The computer is waiting for you to INPUT a value (or assign a value to the variable)

5. Using this program, fill in the boxes:

```
10    INPUT A$
20    INPUT B$
30    PRINT A$, B$
      RUN
      ? THIS
```

A$	This
B$	is easy

6. Review the fact that the values for A$ and B$ variables are not known until you tell the computer what they are when your program is running.

7. Review. When a "?" appears it means that the computer is waiting for you to give it a value. (Interaction)

8. Have class determine the value of each variable.

```
10    INPUT Z$
      RUN
      ? NICE JOB
```

Z$	

```
10    INPUT X$
20    INPUT Y$
      RUN
      ? VERY
      ? GOOD
```

X$	
Y$	

```
10    INPUT A$
20    INPUT B$
30    LET C$ = A$ +  B$
      RUN
      ? GOOD
      ? MORNING
```

A$	GOOD
B$	MORNING
C$	GOOD MORNING

9. Review the INPUT statement cause the computer to wait for a value.

10. Ask what does "?" mean?

CHALLENGE:

11. Using the INPUT statement, write a program that will print the names of 5 students in your class.
12. Write a program that asks what your name is. Use the INPUT statement to tell the computer your name.

UNCONDITIONAL BRANCHING (GOTO)

1. Try this program.

```
10 PRINT "THIS LOOP GOES ON FOREVER"
20 GOTO 10
```

2. Explain that line 20 (GOTO) tells the computer to "go to" the line #10 and continue the program from line 10.

3. Review with the class how to break into a program (CTRL/C or BREAK) then have them run a program similar to the one above.

4. Explain that first the computer executed line 10. Then, it went to line 20. Then back to 10, and back to 20. Illustrate that this will go on "forever" unless we stop it (using CONTROL/C or BREAK, etc.)

5. Explain that because the computer goes on and on "forever" we call this an <u>infinite loop</u>.

6. Ask students to define an infinite loop.

7. Have students create their own program using an infinite loop.

8. Explain that the GOTO statement is a programming device which can be used in more complex programming such as below.

```
10 INPUT "What is your name?";  N$
20 PRINT "Good morning";  N$
30 PRINT "How are you?";  N$
40 PRINT "Goodbye";  N$
50 GOTO 10
```

9. Illustrate that in the above program a user can re-use the program without having to type in RUN each time.

10. Explain that the GOTO statement can also be used in a program to skip over lines.

11. Try this:

```
10 PRINT "I"
20 PRINT "HATE TO"
30 PRINT "EAT ROTTEN FOOD"
40 GOTO 30
```

Now insert line 15.

```
10 PRINT "I"
15 GOTO 30
20 PRINT "HATE TO"
30 PRINT "EAT ROTTEN FOOD"
```

12. Illustrate the path that the computer went through in both programs.

13. Explain that these are examples of unconditional branching.

USING THE GOTO STATEMENT FOR COUNTING

1. Review how to stop or break into a program.

2. Expalin that the students will use an infinte loop again, but this time it will be used to count for us.

3. Illustrate this program:

```
10 LET A = 1
20 PRINT A
30 LET A = A + 1
40 GOTO 20
```

4. Have the students discuss what might happen in the above program.

5. Have the students RUN the program.

6. Change line 20 to PRINT A,

7. Explain how the computer goes through the program.

```
10 LET A = 1
20 PRINT A
30 LET A = A + 1
40 GOTO 20
```

8. Illustrate by tracing the computers path through the program. Trace the value of A.

	1st Time	2nd Time	3rd Time
10 LET A = 1	A = 1		
20 PRINT A	A = 1	A = 2	A = 3
30 LET A = A + 1	A = 2	A = 3	A = 4
40 GOTO 20	A = 2	A = 3	A = 4

9. <u>Assignment</u>

Using an infinite loop write a program that counts by 5.

READ/DATA (using 1 variable)

1. Review the use of INPUT statements. Explain that INPUT statements are used to assign a value to a variable. Review definition of value and variable. Illustrate:

   ```
   10 INPUT "What is your name";  N$
   20 PRINT  N$
   ```

   Explain that the value of N$ is what ever you give the computer. Remind the class that with INPUT statements the computer waits for you to give it a value.

2. Explain that another way to give a value to a variable is to use a READ/DATA statement.

3. Illustrate:

   ```
   10 READ N$
   20 PRINT N$
   30 DATA JOHN
   RUN
   JOHN
   ```

4. Explain that the READ statement forced the computer to look for a DATA statement. When it found the DATA statement it took the value of JOHN and gave it to the variable N$.

5. Explain that it is different from the INPUT statment in that the computer does not wait for the user to give a value to N$. Instead the variable N$ gets its value from the DATA statement.

6. Show how the order of the statements does not matter. Illustrate:

   ```
   10 DATA JOHN          10 READ N$
   20 READ N$            20 DATA John
   30 PRINT N$           30 PRINT N$
   RUN                   RUN
   JOHN                  JOHN
   ```

7. Emphasize for every READ statement there must be a DATA statement.

8. Explain that within DATA statements, "quotes" are not needed - even for strings.

9. Illustrate how the READ/DATA statements work with integers.

```
10 READ B
20 DATA 35
30 PRINT B
RUN
35
```

READ/DATA (using more than 1 variable)

1. Review the READ/DATA statements. Point out that they are used to assign values to variables.

   Also, if a program has a READ statement then it must also have a DATA statment.

   Also, the order of the READ/DATA statements does not matter.

2. Explain that the READ/DATA statement can be used to assign values to many variables.

3. Illustrate:

```
10 READ A$, B$, C$
20 PRINT A$, B$, C$
30 DATA I, AM, GREAT
```

4. Explain that in this program, the value "I" is assigned to A$, the value "AM" is assigned to B$, and the value "GREAT" is assigned to C$.

5. Explain that " " are not needed in DATA statements and that commas are needed between values.

6. Explain that a comma is not used after the last value.

7. Explain that the PRINT statement can be changed to produce a different result.

8. Illustrate:

```
10 READ A$, B$, C$
20 PRINT B$, A$, C$
30 DATA I, AM, GREAT
RUN

AM I GREAT
```

9. Review:

   - Use of commas in the DATA statement.

   - Manipulation of PRINT statement to change output.

IF/THEN Conditional Branching

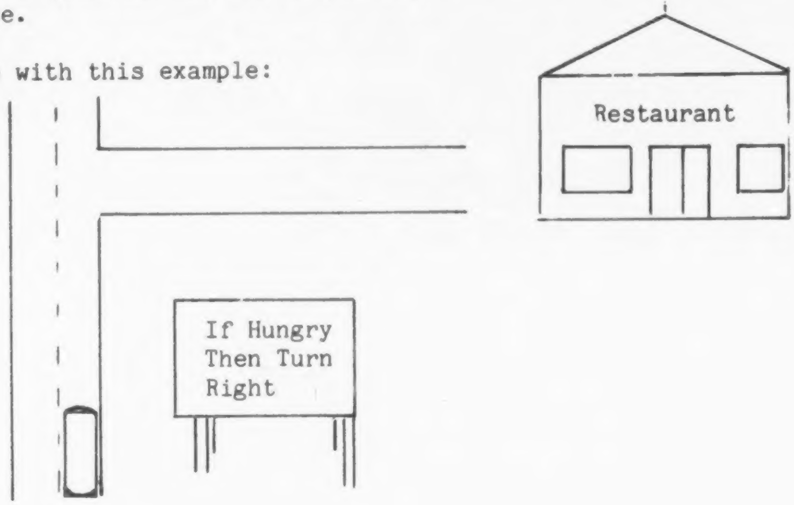
1. Explain that branching is what happens when the computer follows an instruction that makes it go to another section of the program.

   It does not follow the order of the line numbers when it branches.

   A GOTO statement is a branching statement.

2. The difference between GOTO statments and conditional branching statemnts is that the computer goes to different parts of the program only if a particular condition is true or not true.

   Illustrate with this example:



   In this illustration, if the driver of the car is hungry, he/she will turn right and go to the restaurant.

   If he/she is not hungry he/she will continue on the road.

3A. The computer works in a similar way. If a condition is true the computer will branch to a different part of the program.

   If the condition is not true the computer will follow the line number order.

4. Explain when using computers we use an IF/THEN statement.

5. Illustrate:

Line # _____ IF (Condition is true or not true) THEN (Action)

20 IF driver is hungry, THEN turn right

Explain

6. When using a BASIC statment it will look like:

```
20 IF x = 5 THEN GOTO 30
```

7. Explain that the IF/THEN statement is useful also for telling the computer to print a message when a certain condition is true or not true.

8. Explain how to make the computer print a message. Use the following format.

Line # IF (Condition) THEN PRINT (anything you want)

9. A BASIC IF/THEN statment will look like:

```
20 IF x = 0 THEN PRINT 'You have a $0 balance in
your account."
```

Appendix C

Instructions for Computer Career Research Project

(and Metroguide printout)

for use with Computer Career Lessons

Purpose: Students will obtain more extensive and explicit information about careers in the computer field by performing hands-on library research into one of the careers discussed in the classroom lessons.

IMPLEMENTATIONS: Students will be required to do the following:

1. Select any one of the careers discussed in classroom lessons or a similar career in the computer field.
2. Investigate the selected career in detail by conducting research using one or more of the following library sources:
   a. Dictionary of Occupational Titles
   b. Occupational Outlook Handbook
   c. Other career-oriented publications available form school library, public library, guidance personnel, State Employment Counsellor, etc.
3. Student's report (either written or oral) should demonstrate that research was made into the following areas of investigation:
   a. Title of career area
   b. Requirements for employment
   c. How training may be obtained (schools, on-job, etc.)
   d. What one would expect to do in a typical day at work in this particular career (specific job duties.)
   e. Opportunities for advancement from this position to higher rungs on career ladder.
   f. Most recent information about financial reward (salaries, fringe benefits, etc.)
   g. Informed evaluation of career: Now that student has obtained more information about career area, have his/her opinions been revised? Why or why not?

4. Students should be encouraged to read help-wanted advertisements for the career of interest and mount them as part of the completed report.

5. Students should also be encouraged to obtain firsthand information by writing letters to various data-processing organizations requesting additional details about the career of interest. Addresses of these organizations can be obtained from librarians.

Metro Guide Exerpts

Metro Guide is a computer-based career counseling service available in selected high schools. The following excerpts are printouts of computer careers.

Metroguide Printout

```
DESC 2354
 2354 COMPUTER PROGRAMMERS     (3/82)
COMPUTER PROGRAMMERS WRITE COMPUTER PROGRAMS AND OTHER CODED
INSTRUCTIONS FOR COMPUTERS WHICH ENABLE THEM TO PERFORM A DESIRED
TASK.  THEY FIRST DECIDE WHAT INFORMATION WILL BE NEEDED TO SOLVE
A PROBLEM.  NEXT, THEY PREPARE A FLOWCHART OR DESCRIPTION TO OUT-
LINE THE MACHINE OPERATIONS IN GENERAL TERMS.  USING THE FLOW-
CHART, THEY WRITE COMPLETE, STEP-BY-STEP INSTRUCTIONS FOR THE
COMPUTER.  THE PROGRAM IS THEN TESTED, ERRORS ARE CORRECTED,
AND PROCEDURES ARE RECORDED.  SPECIALTIES:  BUSINESS PROGRAMMERS
(DOT# 020.162-014), ENGINEERING AND SCIENTIFIC PROGRAMMERS (020.
167-022),, INFORMATION SYSTEM PROGRAMMERS (020.187-010), PROCESS
CONTROL PROGRAMMERS (020.187-014).  CIS RELATED OCCUPATIONS:
SYSTEMS ANALYSTS (SEE DESC 2355), COMPUTER OPERATORS (2356).
---APTITUDES:  ABILITY TO ORGANIZE IDEAS AND DATA, ABOVE AVERAGE
NUMERICAL ABILITY AND CLERICAL PERCEPTION.
---WORK SETTING:  USUALLY IN OFFICES WITH SOME WORK DONE IN COM-
PUTER ROOMS.  SOME NIGHT AND WEEKEND WORK MAY BE NECESSARY WHEN
NEW PROGRAMS ARE TESTED OR QUICK CHANGES ARE BEING MADE.  EM-
PLOYERS:  WHOLESALE AND RETAIL BUSINESSES, BANKING AND IN-
SURANCE FIRMS, GOVERNMENT AGENCIES, ELECTRONICS AND OTHER MANU-
FACTURERS, TRANSPORTATION AND PUBLIC UTILITIES, RESEARCH
ORGANIZATIONS, COLLEGES AND PUBLIC SCHOOL SYSTEMS.
---HIRING PRACTICES:  AT LEAST ONE YEAR OF FORMAL TRAINING IS
PREFERRED FOR BUSINESS PROGRAMMERS.  MOST ENGINEERING AND SCIEN-
TIFIC PROGRAMMER POSITIONS REQUIRE FOUR YEARS OF FORMAL TRAINING.
(SEE PREP 2354).
---CURRENT EMPLOYMENT:  AROUND 15,000 IN THIS MEDIUM-SIZED OCCU-
PATION IN NEW YORK CITY.
---WAGES:  AVERAGE ENTRY RATE IN NEW YORK CITY IS $1,500/MONTH.
MOST EARN BETWEEN $1,900-2,550/MONTH.  SENIOR PROGRAMMERS EARN
BETWEEN $2,200-2,675/MONTH.  COMPUTER PROGRAMMERS IN NASSAU
AND SUFFOLK COUNTIES BEGIN AT $800/MO.  MOST EARN ABOUT $1,250/MO
THE AVERAGE MAXIMUM IN NASSAU/SUFFOLK IS $1,600/MO.
---OUTLOOK:  SHORTAGE OF PROGRAMMERS IN NEW YORK CITY AND THROUGH
OUT THE STATE.  OUTLOOK DEPENDS ON EXPANDED USE OF COMPUTERS IN
BUSINESS, GOVERNMENT, MEDICINE AND EDUCATION, AS WELL AS TECH-
NOLOGICAL ADVANCES IN COMPUTER HARDWARE.  CURRENTLY, DEMAND IS VE
HIGH IN NEW YORK CITY AS WELL AS IN THE SURROUNDING NEW YORK MET-
ROPOLITAN AREA.  DEMAND IS EXPECTED TO REMAIN HIGH THROUGHOUT THE
1980'S AND INTO 1990.  WITH THE ADVENT OF OFFICE-OF-THE-FUTURE
TECHNOLOGICAL DEVELOPMENTS SUCH AS MICROCOMPUTERS, AND WITH THE
GROWING COMPLEXITY OF BUSINESS RECORD KEEPING, AN INCREASING NUM-
BER OF BUSINESSES ARE PURCHASING DATA PROCESSING EQUIPMENT TO PRO
CESS LARGE VOLUMES OF CLERICAL AND ACCOUNTING DATA.  WHILE THIS
MAY DECREASE THE NUMBER OF PROGRAMMERS NEEDED BY DATA PROCESSING
SERVICE FIRMS, BUSINESSES WILL CONTINUE TO HIRE PROGRAMMERS TO
ADAPT BUSINESS OPERATIONS TO SMALL COMPUTERS.
```

PREP 2354

2354 COMPUTER PROGRAMMERS (3/82)

---SKILLS: KNOWLEDGE OF PROGRAMMING TECHNIQUES AND LIMITATIONS, ABILITY TO ANALYZE DATA AND WORK WITH LITTLE ROOM FOR ERROR. ABILITY TO COMMUNICATE TECHNICAL INFORMATION, BOTH ORALLY AND IN WRITING, AND TO ORGANIZE TIME IN ORDER TO MEET DEADLINES.

---PREPARATION: THE LEVEL OF TRAINING NECESSARY VARIES GREATLY WITH THE TYPE OF COMPUTER PROGRAMMING. TRAINING MAY VARY IN LENGTH FROM A 20 WEEK COURSE TO MASTER'S AND DOCTORAL PROGRAMS IN COMPUTER SCIENCE OR ENGINEERING. BUSINESS APPLICATIONS PROGRAMMERS USUALLY HAVE ONE OR TWO YEARS OF TRAINING IN PROGRAMMING SUPPORTED BY TRAINING OR EXPERIENCE IN ACCOUNTING, BUSINESS ADMINISTRATION OR MATH. SCIENTIFIC OR ENGINEERING APPLICATIONS PROGRAMMERS USUALLY HAVE A BACHELOR'S DEGREE IN COMPUTER SCIENCE, MATHEMATICS, ENGINEERING OR THE PHYSICAL SCIENCES. A GRADUATE DEGREE MAY BE REQUIRED FOR SOME SCIENTIFIC OR ENGINEERING SYSTEMS PROGRAMMER POSITIONS.

RELATED EDUCATIONAL PROGRAM: DATA PROCESSING (SEE PROG 211).

---TIPS: EMPLOYERS PREFER APPLICANTS WHO HAVE TRAINING IN A FIELD RELATED TO THEIR APPLICATION, SUCH AS BUSINESS ADMINISTRATION, PHYSICAL SCIENCE OR ENGINEERING. ALSO, EXPERIENCE ON THE MODEL OR TYPE OF COMPUTER THE EMPLOYER USES IS VERY HELPFUL.

DESC 2356

2356 COMPUTER OPERATORS (3/82)

COMPUTER OPERATORS MONITOR AND CONTROL COMPUTERS TO PROCESS DATA ACCORDING TO OPERATING INSTRUCTIONS. DUTIES INCLUDE SELECTING AND LOADING INPUT AND OUTPUT UNITS WITH MATERIALS SUCH AS TAPES, PUNCH CARDS OR PRINTOUT FORMS, AND OBSERVING THE MACHINES FOR STOPPAGE OR FAULTY OUTPUT. SPECIALTIES: COMPUTER OPERATORS (DOT# 213.362-010), COMPUTER PERIPHERAL EQUIPMENT OPERATORS (213.382-010). CIS RELATED OCCUPATIONS: DATA ENTRY OPERATORS (SEE DESC 1438), SYSTEMS ANALYSTS (2353), & PROGRAMMERS (2354).

---APTITUDES: TYPING ABILITY AND METICULOUS APPROACH TO PROBLEMS.

---WORK SETTING: WORK IN WELL-LIGHTED, CLEAN BUT NOISY AREAS IN SOMEWHAT COOL TEMPERATURES (65 DEGREES); WALK AND STAND FOR LONG PERIODS ON CONCRETE FLOORING; HOURS MAY INCLUDE EVENING, WEEKEND AND NIGHT DUTY. EMPLOYERS: RESEARCH, BANKING, INSURANCE, WHOLESALE & RETAIL, TRANSPORTATION & PUBLIC UTILITY FIRMS; GOVERNMENT AGENCIES; PUBLIC SCHOOL SYSTEMS.

---HIRING PRACTICES: ADVANCED TRAINING BEYOND HIGH SCHOOL IS USUALLY SPECIFIED. HOWEVER, A HIGH SCHOOL GRADUATE SHOWING AN INTEREST AND APTITUDE FOR THE WORK MAY RECEIVE ON-THE-JOB TRAINING IN SMALLER DATA-PROCESSING FIRMS. TRAINING: AVAILABLE AT COMMUNITY COLLEGES AND PRIVATE SCHOOLS AS WELL AS IN SOME NEW YORK CITY HIGH SCHOOLS; SOME ACQUIRE SKILLS ON THE JOB. SEE PREP 2356 FOR DETAILS.

---CURRENT EMPLOYMENT: AROUND 15,500 EMPLOYED IN THIS MEDIUM-SIZED OCCUPATION IN NEW YORK CITY.

---WAGES: AVERAGE ENTRY RATE IN NEW YORK CITY IS AROUND $900/ MONTH. MOST EARN BETWEEN $1,050-1,200/MONTH. COMPUTER OPERATORS BEGIN AT 640/MO IN NASSAU AND SUFFOLK COUNTIES. MOST EAR ABOUT $870/MO. THE AVERAGE MAXIMUM WAGE IN NASSAU/SUFFOLK IS $1,360/MO.

---OUTLOOK: BALANCE TO SLIGHT SHORTAGE IN NEW YORK CITY. OUTLOOK DEPENDS ON EXPANDED USE OF COMPUTERS AND TECHNOLOGICAL ADVANCES IN ELECTRONIC DATA PROCESSING EQUIPMENT AND METHODS. EMPLOYMENT OF COMPUTER OPERATORS IS EXPECTED TO RISE ABOUT AS FAST AS THE AVERAGE FOR ALL OCCUPATIONS. OPPORTUNITIES ARE SLIGHTLY BETTER FOR THOSE WITH EXPERIENCE OR THE WILLINGNESS TO WORK EVENING-WEEKEND SHIFTS. THE DEMAND FOR COMPUTER OPERATORS IS INCREASING IN NEW YORK CITY AND THROUGHOUT THE STATE.

BEST COPY AVAILABLE

PREP 2356
2356 COMPUTER OPERATORS (3/82)

---SKILLS: KNOWLEDGE OF COMPUTER SYSTEMS AND EQUIPMENT, TYPING PROFICIENCY, FAMILIARITY WITH TECHNICAL LANGUAGE USED IN OPERATING INSTRUCTIONS AND COMPUTER MANUALS, AND ABILITY TO FOLLOW DETAILED AND ORGANIZED PROCEDURES.

---PREPARATION: ADVANCED TRAINING BEYOND HIGH SCHOOL IS USUALLY SPECIFIED FOR COMPUTER OPERATORS. TRAINING IS AVAILABLE IN PRIVATE VOCATIONAL SCHOOLS, COMMUNITY COLLEGES AND UNIVERSITIES. A HIGH SCHOOL GRADUATE WITH INTEREST AND APTITUDE FOR THE WORK MAY RECEIVE ON-THE-JOB TRAINING IN SMALL DATA PROCESSING FIRMS.

RELATED EDUCATIONAL PROGRAM: DATA PROCESSING PROGRAMS (SEE PROG 211).

DESC 1438
1438 DATA ENTRY OPERATORS (3/82)

DATA ENTRY OPERATORS USE MACHINES SIMILAR TO TYPEWRITERS TO TRANSCRIBE INFORMATION INTO A FORM THAT A COMPUTER CAN READ. THEIR DATA ENTRY MACHINES MAY MAKE PATTERNS OF HOLES REPRESENTING LETTERS AND NUMBERS ON KEYPUNCH CARDS. SOME RECORD THE DATA ON MAGNETIC TAPE OR DISKS; OTHERS ENTER THE DATA DIRECTLY INTO THE COMPUTER. SOME DATA TERMINALS HAVE A TELEVISION-LIKE SCREEN (CRT'S) THAT DISPLAYS THE DATA AND PERMITS CORRECTIONS BEFORE IT IS ENTERED. SPECIALITIES: KEYPUNCH OPERATORS (DOT #203.582-030); DATA-CODER OPERATORS (203.582-026); TERMINAL OPERATORS (203.582-054); DATA TYPISTS (203.582-022). CIS RELATED OCCUPATIONS: COMPUTER OPERATORS (SEE DESC 2356), OFFICE MACHINE OPERATORS (1439), CLERK TYPISTS (1416).

---APTITUDES: ABILITY TO WORK QUICKLY AND ACCURATELY FOLLOWING SPECIFIC PROCEDURES; GOOD EYE-HAND COORDINATION; GOOD VISION; AND ABLE TO WORK UNDER CLOSE SUPERVISION SOMETIMES AS PART OF A TEAM. MUST LIKE WORKING WITH MACHINES.

---WORK SETTING: USUALLY IN AN OFFICE. DATA ENTRY OPERATORS MAY MAY BE EXPOSED TO NOISY SURROUNDINGS. EMPLOYERS: MOST WORK FOR MANUFACTURERS, WHOLESALE AND RETAIL COMPANIES; BANKS; GOVERNMENT AGENCIES; INSURANCE COMPANIES AND DATA PROCESSING FIRMS.

---HIRING PRACTICES: GRADUATION FROM HIGH SCHOOL OR BUSINESS SCHOOL. TRAINING: AVAILABLE AT HIGH SCHOOLS, PRIVATE BUSINESS SCHOOLS AND COMMUNITY COLLEGES (SEE PREP 1438).

---CURRENT EMPLOYMENT: AROUND 21,200 ARE EMPLOYED IN NEW YORK CITY, A LARGE-SIZED OCCUPATION.

---WAGES: THE AVERAGE ENTRY WAGE IN NEW YORK CITY IS ABOUT $700/MONTH. THOSE WITH EXPERIENCE EARN ABOUT $900/MONTH. THE AVERAGE MAXIMUM IS ABOUT $1,100/MO IN NEW YORK CITY. DATA ENTRY OPERATORS IN NASSAU AND SUFFOLK COUNTY BEGIN AT $600/MO. THOSE WITH EXPERIENCE EARN $770/MO. THE AVERAGE MAXIMUM IS $1,300/MO IN NASSAU/SUFFOLK.

---OUTLOOK: BALANCE TO SLIGHT SURPLUS. AS THE TECHNOLOGY DEVELOPS, CONTINUED GROWTH IS EXPECTED IN THE USE OF COMPUTERS, ESPECIALLY BY SMALL BUSINESSES. THEREFORE, DEMAND FOR DATA ENTRY OPERATORS (CONSOLE AND PERIPHERAL EQUIPMENT OPERATORS) IS EXPECTED TO INCREASE ABOUT AS FAST AS THE AVERAGE FOR ALL OCCUPATIONS THROUGH THE 1980'S WHILE THE DEMAND FOR KEYPUNCH OPERATORS SHOULD CONTINUE TO DECLINE. DEMAND FOR DATA ENTRY OPERATORS IS CURRENTLY HIGH IN NEW YORK CITY AND IN WESTCHESTER COUNTY. PROSPECTS ARE BEST FOR APPLICANTS WHO ARE FAMILIAR WITH SEVERAL FORMS OF DATA ENTRY WHO ARE WILLING TO WORK EVENINGS, NIGHTS, AND SPLIT SHIFTS.

PREP 1438

1438 DATA ENTRY OPERATORS (3/82)

---SKILLS: ABLE TO TYPE 30-40 WORDS PER MINUTE, WORK WELL WITH DETAIL, AND FOLLOW INSTRUCTIONS; FAMILIARITY WITH DATA ENTRY EQUIPMENT SUCH AS CRT'S AND KEYPUNCHING MACHINES. A BASIC KNOWLEDGE OF DATA PROCESSING INCLUDING AN UNDERSTANDING OF SPECIAL VOCABULARY IS HELPFUL.

---PREPARATION: MINIMUM EDUCATION REQUIRED IS HIGH SCHOOL GRADUATION. MOST EMPLOYERS REQUIRE SOME POST-HIGH SCHOOL TRAINING OR EXPERIENCE FOR DATA ENTRY OPERATORS. TRAINING IS AVAILABLE AT PRIVATE VOCATIONAL SCHOOLS AND COMMUNITY COLLEGES. SOME OPERATORS BEGIN AS CLERKS AND LEARN DATA ENTRY ON THE JOB.

RELATED EDUCATIONAL PROGRAM: KEYPUNCH AND DATA ENTRY (SEE PROG 214).

Appendix D

Use of Peripheral Devices

NOTE TO THE TEACHER: This section is a mini-resource unit, the topics of which may be used as part of other lessons in this guide at a time you deem to be appropriate. How you use parts of this resource material depends on your own school setting and especially on the quality and quantity of peripheral devices available to you. It is, of course, impossible to cover all computers and peripherals here. However, an introduction to the major peripherals used for storage and hardcopy is provided here, as well as a discussion, in very general terms, of how they work with some of the more popular micros, such as the Apple II, Atari 400/800, Commodore Pet, IBM PC and Radio Shack TRS-80 with Level II Basic.

STORAGE DEVICES:

- The Cassette Recorder and Tapes.
  - How to Use:
    1. The cassette recorder is connected to the computer as described in the computer user's manual.
       - When a student records a program from the computer's memory onto his or her cassette tape, he or she is saving his or her program. (As the tape, coated with ferric oxide passes the record/play head, the program is saved on the cassette magnetically as a series of high-pitched sounds.)
       - When he or she plays the program back from the tape into the computer's memory, he or she is loading his or her program.
    2. To keep track of what is on a tape, the students must make a list of all programs that they have recorded. (Since tapes are relatively inexpensive, each student can have his or her own cassette to record programs.)
       - If the machine has a counter, the students should also record the starting and ending number of each program.

    - If the recorder does not have a counter, some other method (e.g., markings on the window of the cassette itself) should be used to help determine the position of a program on the tape.

  3. Students must advance the tape after each program.(They must not record programs too closely together.)

      - This avoids over-writing a previously recorded program since the computer cannot check to see if a blank tape is being used.

  4. Students should save their programs more than once.

      - This practice will help to make sure students have a good recording of each program.

  5. To load a program into the computer from cassette, the Apple and PET use the LOAD command, the IBM uses LOAD (CAS1:), and the Atari and TRS-80 use CLOAD.

      - To save a program to cassette, the Apple and PET use the SAVE command, the IBM uses SAVE (CAS1:), and the TRS-80 and Atari use CSAVE.

- Advantages:

  1. Most computers will work with a cassette recorder.

  2. The cassette is used as a storage device because it is relatively inexpensive. A cassette recorder and tape cassettes are less expensive than disk drives and disks.

  3. Regular audio tapes can be used. Purchase shorter length cassettes by quality manufacturers or special cassettes made for computers.

- Disadvantages:

  1. Tapes are slow to record information from and play back into the computer.

2. Since tapes (especially cheaper ones) have been known to break, stretch, drop signals, etc., the students should make a backup copy (preferably on another tape) of any important program.

- Care and Feeding:
  1. TAPE MACHINE:
     - Keep foreign matter out of the machine.
     - Do not touch either the tape or the record/play and erase heads.
     - Clean the heads with a head cleaning cassette.
     - Insert the tapes carefully and correctly to prevent damaging the machine.
     - Keep the recorder covered and dust-free.
  2. TAPES:
     - Store cassette tapes in their cases when not in use.
     - Keep in a dry spot out of the sun.

- The Disk Drive and Diskettes.
  - How to Use:
    1. Programs may be stored on floppy disks (a.k.a. diskettes, disks, and floppies), which look something like 45 R.P.M. records. They are thin disks coated with ferric oxide and housed permanently in protective envelopes.
       - A floppy disk, which is either 5.25 or 8 inches in diameter, still wearing its protective sleeve, is placed inside a disk drive, which may be built into the computer or connected to it.
       - The disk spins rapidly inside the drive. A read/write head moves across its surface through a slot in the protective covering.
       - The head reads (or loads) data from the disk and writes (or saves) data to it.

2. Each computer has its own method for making a disk ready to accept data (format) and a method of inserting a disk into a drive.

   - If the disk is inserted improperly, the drive or the disk could be damamged.

   - Do not open the door of the drive when the "in use" light is lit.

3. If students save their programs on their own disks, they should write their names on the diskette label with a soft tipped pen so they don't damage the disk.

4. To LOAD a program from disk or SAVE a program to disk, on the Apple or IBM, they would type LOAD "FILENAME", on the PET type DLOAD "FILENAME", and on the Atari type LOAD "D:FILENAME".

   - The command formats are identical for saving data to disk, except the word SAVE must be substituted for LOAD.

- Advantages:

1. Disk drives are very easy to use once the proper commands are learned. Disk drives SAVE and LOAD programs much faster than tape recorders. They are also more reliable than cassettes.

2. The disk automatically keeps a list (called a catalog or directory) of all programs that have been saved to it. Every program is saved with a name so the computer can easily set aside a place on the disk to store it and, subsequently, find it. Unlike working with the tape, you don't have to worry about the program's location on the diskette.

3. A disk can save more programs on it than a tape can.

- Disadvantages:

1. Diskettes are more expensive to buy than tapes and can only be bought at stores that specialize in computers and accessories for computers.

2. Each computer has a different way of formatting a disk. Formatting a new blank disk must be done before the computer can read or write to it.

- Care and Feeding:

  1. DISK DRIVE:

     - Do not to insert any foreign matter into the drive.
     - Keep the drive covered when not in use so dirt and dust don't get into the machine.
     - Follow the correct start-up procedures for that particular computer.
     - When the drive is being used, any ventilation holes must not be covered or the drive might heat up.
     - The read/write head of the drive must be kept clean. This can most easily be accomplished through the use of "head cleaning" diskettes.

  2. DISKETTES:

     - Though they are called "floppies," they can be ruined if they are bent.
     - When they are not being used they must be placed in their protective outer envelopes.
     - Store in a dry place away from excessive heat or any magnetic field.
     - Touch only the jacket and not the magnetic recording surface of the disk itself.
     - Since it is possible to ruin a disk, the students should make backups of their programs on a second disk.

- The Printer.

  - How to Use:

    1. A printer is capable of producing hardcopy, or printout, of program listings and other data.

  - Every character (number, letter, etc.) travels from the computer to the printer through a cable and interface. Signals travel in the form of a numerical code called the American Standard Code for Information Interchange (ASCII).

2. There are two popular types of printers, dot matrix and daisy wheel:

    - The dot matrix printer has a printing head consisting of a number of wires, which pop out and make "dots" on the paper in the form of letters, numbers and other characters.

    - The daisy wheel printer types one full character after another as a typewriter does.

3. You may want your students to use a printer to obtain a hardcopy of their program listings. For the IBM and TRS-80, type LLIST. For the Apple, first type PR#1 and then type LIST. For the PET, first type OPEN1,4 and then CMD1 and finally LIST.

- Advantages:

  - Dot Matrix Printer:

    1. It is less expensive than a daisy wheel.

    2. It is capable of doing graphics.

  - Daisy Wheel Printer:

    1. Print is letter quality and can be used for important correspondence.

    2. As with certain typewriters, the user can change the type fonts for different print styles and sizes.

- Disadvantages:

  - Dot Matrix Printers:

    1. Since these printers do their work with wires that come out and strike the paper, the letters are made up of dots. The

density of the dots depends on the printer: The more dots per letter, the nicer the print is.

2. Some computer-phobes don't like receiving correspondence produced on a dot matrix printer, no matter how dense the dots.

- Daisy Wheel Printers:

    1. The daisy wheel printer is usually slower and noisier than dot matrix.

    2. It is more expensive to buy.

- Care and Feeding:

    1. Check that the printer has paper and that the ribbon is in good condition. There must be no obstacle to interfere with the flow of paper.

    2. The ribbon should be checked periodically to make sure that it hasn't stretched or come out of its holder and that it is still sufficiently inked.

    . Paper should be kept in the printer at all times, and the printer should be kept covered when not in use.

SUGGESTED APPLICATIONS OF PERIPHERALS:

- Have the students write a sample program. Then have them SAVE it to cassette and/or disk, erase it from memory, and then LOAD it again. Also have them produce a hardcopy of the listing of their program on a printer.

- Using the information contained in the How to Use sections of this appendix, other reference sections of this curriculum guide and computer manuals, develop the following chart.

COMMANDS FOR POPULAR MICROCOMPUTERS

FUNCTION	APPLE	ATARI	PET	IBM-PC	TRS-80
Load from Cassette					
Save to Cassette					
Load from Disk					
Save to Disk					
List to Printer					

- Have members of the class research how other computers (e.g., Sinclair ZX81; VIC20; Texas Instruments 99/4A; TRS-80 Color Computer; Commodore 64; etc.) communicate with storage devices and printers.

Appendix E

Computer Selection and Care

In recent years the microcomputer has found its way into offices, homes, and schools in increasing numbers. Manufacturers, retailers, and educators are telling people everywhere of the advantages of computers, and how they can transform the home, school, and office into a more efficient place. The following information is for your consideration when setting up a computer room in your school or home. This section will cover the setup and maintenance of the microcomputer you have chosen, regardless of manufacturer or model.

A NOTE ABOUT POSSIBLE HAZARDS

There has been some mention in literature recently about the possible health hazards of radiation from the screen of some computers or video display terminals. The National Institute of Occupational Safety and Health (NIOSH) has done preliminary research into possible hazards and has assured us that the danger is virtually nonexistent.

The findings of the NIOSH survey are that, in most cases, the radiation in the X-ray, ultraviolet, and infrared spectra are safe, frequently being below the level that present instruments can detect. Modern computers, monitors, and television receivers are all made in conformity with Federal Communications Commission specifications which limit the amount of radiation from equipment to levels considered safe for long exposures. The only hazard they foresee occurs when a computer is attached to a color television set manufactured prior to January, 1971. Sets manufactured before that date did not have the same specifications, and some of them did emit X-rays from their high voltage stages. Most of these sets have, however, outlived their useful lives, and have been discarded. If you are using, or are planning to use a computer with a color television set as a display, check the date of the manufacture. This is usually printed on the back of the set.

The researchers at NIOSH investigated other problems associated with the use of microcomputers and VDTs. These health problems are much less dangerous than the radiation hazard and, fortunately, are very easy to prevent.

Eye strain seems to be the major problem operators associate with computers and VDTs. Since the face of the screen is usually smooth and shiny, it reflects light from the surrounding room, from windows, light fixtures, and the operator's clothing. A rapidly strobing television picture tube can also produce problems, as can a display in which contrast and brightness are improperly adjusted.

Another problem found by the NIOSH researchers is posture problems leading to back and shoulder pains. Care should be taken, when setting up a computer room, to ensure that equipment is placed to prevent both vision and posture problems for the users.

SETTING UP

Furniture is a major consideration when setting up the computers. The right furniture can make computing most pleasurable. The wrong furniture can make it unbearable.

Both keyboard and display levels are crucial. Keyboards should be kept at a level comfortable for the user. Young children, particularly, should not have the keyboard mounted so high that they have to reach up at an uncomfortable angle to type. Using a school desk with the top removed, and placing the computer on the book tray, is one way to keep the equipment at a proper level. If your computer has a built-in monitor, like the TRS-80 or the PET, this will also keep the screen low enough to reduce reflections from the ceiling lights and other similar sources. The height of the screen should be considered, regardless of the type of computer, to reduce glare and to prevent the user from having to look up too high, causing neck pains. Check to see that the mounting table is sturdy, so it will not tip over or tilt allowing the computer to fall off accidentally.

Another consideration is the stationing of equipment relative to windows and doors. The video tube will reflect the light coming through windows and doors. Care should be taken to keep the monitor positioned so that this is minimized. Do not, however, place the computers so that the user has to face the source of light, as that can also cause eyestrain.

MONITOR OR TELEVISION - WHICH TO USE

Some computers have built in screens. Others, like the Apple II, the Atari, the Commodore VIC and 64, and the Texas Instruments TI 99/4A need a separate display. These units may be used with either a television or a video monitor. While a monitor outwardly resembles a television, there are several important differences. The choice of a television depends on several factors. Here are some facts which may help you decide.

A video monitor is basically a television set that is built without a tuner, or channel selector. It is designed to be attached directly to a computer or other video source. Because a monitor is not used for television use, it can be designed to be more efficient than a television for computer use.

A television picture is actually "drawn" 30 times a second by an electron beam. The beam strikes a phosphor on the inside of the screen, and makes it light up. Since the objects shown on the screen are usually moving, often quite rapidly, the picture must be completely redrawn each time. When used with a computer, the television also redraws the picture 30 times a second. In this case, however, there is really no need for this, as the text on the screen does not move. The redrawing of the picture makes the tube flicker, or "strobe", 30 times a second. Prolonged use of the computer can cause headaches or eyestrain under these conditions.

Because the computer must "modulate" the signal, or turn it into a miniature broadcast, and then have the television turn it back into a demodulated signal, the use of a television set produces a picture that contains considerable interference. Because of this, computers designed for use with television sets have narrower displays. The Apple II and Ataris have only 40 characters across the screen. Other brands have fewer. The TRS-80, with its monitor, has 64 "columns." The Commodore CBM, the business version of the PET, has 80 columns. This is because the monitor has inherently better resolution since there is no need to convert a broadcast signal back to a picture signal.

Monitors also usually have a "slow-decay" phosphor in the picture tube. This phosphor glows longer after being lit by the electron beam. Usually, the dots stay lit until the beam strikes them again. This prevents the strobing that comes with use of a television set. Green or amber phosphor tubes are also available in monitors, as are anti-glare surfaces on the tubes. These further reduce eyestrain. For approximately $100.00, you can purchase a good non-color monitor.

CARING FOR YOUR COMPUTER

A microcomputer is basically a durable, reliable unit. It does not require the controlled environment that a large mainframe unit does. However, it does need some care and maintenance to ensure long and faithful service.

Do not allow eating around the computer. This rule should be strictly enforced in all computer rooms. Crumbs fall into the keyboard and interfere with the contacts, rendering various keys useless, and requiring expensive repairs. Salty foods, such as potato chips, corn chips, etc., are the worst offenders because the crumbs stick unseen to the fingers and, when they drop into the keyboard, the salt causes corrosion. Liquids, such as soda, coffee, etc., are just as bad. A spill into the computer usually turns into a trip to the shop and, at today's rates, the repair can often cost over $100. Keep all food away.

Smoking is prohibited - even at home. Smoke contains substances that can cause corrosion of the contacts and the printed circuits that make up the circuit board. Shorts and broken circuits result, necessitating repairs. Smoke particles are just as bad for tape and disk drives because these can cause the heads to lose contact with the magnetic coating, and miss some of the information stored there. These are also abrasive and can wear away the coating, causing permanent loss of data.

Keep the computer covered. Whenever your computer is in use, the heat created causes convection currents to circulate. These currents cool the computer, but also bring in dust. While some dust will enter while the computer is in use, the entry of dust when the computer is off is very easy to prevent. Just keep the computer covered.

Computer covers are inexpensive, and return their cost in savings on repairs. Radio Shack makes covers for all of its computers and peripherals. Most sell for five or six dollars. Many other manufacturers also have plastic covers for various equipment. If your computer manufacturer has them, buy them. If not, they are available from many aftermarket suppliers. Covers are a "must have" item.

Don't try to make your own dust covers! These items are made of special materials, treated to be static free. Static electricity is even more harmful to computers than dust. One discharge can wipe out the entire computer! No old shower curtains or plastic tablecloths. Spend the money; the price is a small enough investment to protect your computer.

Most of us know static electricity through the shock given when you scuff across a rug. One can build up a charge of several thousand volts (yes, thousands) walking across carpeting. If your computer room floor is carpeted, there are certain steps to take to protect your equipment.

Use an anti-static spray. There are a few such sprays on the market. Many sprays sell for about six dollars. Use it at least once a week, especially in the winter, when humidity is low.

Anti-static mats are helpful. These mats go under the computer, and help drain away the static charge from the user when he or she starts to work. Some mats are made for the floor, but these are expensive.

Check proper electrical wiring. There are some computers that have a "hot chassis." That means that the frame of the computer is connected to one side of the power line. It is possible for these computers to pose a serious shock hazard if the room wiring is not done correctly, or if adapters are used to convert three-prong plugs to two-prong outlets. If any shock hazard is detected, UNPLUG YOUR COMPUTER. DO NOT USE THE COMPUTER UNTIL REPAIRS ARE MADE or you will cause damage to the computer; or worse, to the person using it.

It is advisable to have a "dedicated" electric line for the computers. This is a line that goes only to the computer, preventing it from picking up interference from other electrical appliances or lights. Incidentally, flourescent lights can cause all sorts of problems if they are on the same circuit as the computers. Most schools are wired so that lights and outlets are on separate circuits. It is at home that this problem will occur most often.

Do not use adapters. If your computer has a three-prong plug, it should be plugged into a three-prong outlet. Using an adapter can result in the polarity being reversed. If your computer is built with a "hot chassis," this may be a shock hazard or may result in blown fuses. Have a competent electrician install the proper receptacle to be sure that your equipment, and you, have a long life.

It is also suggested that the installation include a master power switch enabling the teacher to shut off the current to all computers and peripheral devices.

Maintenance keeps the computer working. From time to time, a computer and its peripherals require some help to keep them running properly. Diskette and cassette drives pick up a layer of dirt on their read/write heads that cause these to malfunction. Other adjustments go awry and have to be redone. Dirt covers the case and screen. Careful attention can prolong the interval between repairs.

Clean the computer regularly. Although you are using dust covers, the computer will still get dirty. The case can be cleaned with a glass cleaner or mild detergent and a clean, soft cloth. Spray the cleaner onto the cloth, NEVER onto the computer's case. The spray will find its way inside and cause corrosion of the circuits. The screen can be cleaned this way also, but it won't last long. The screen gets a static charge from the electron beam, and this attracts dust. Spray your anti-static spray onto a cloth and gently wipe the screen with that. The cloth should be damp, not wet, or the liquid will get past the edge of the tube and into the set.

Clean your cassette drive heads frequently. This can be done with head cleaning solution and a cotton-tip swab. If you can't get this cleaner, use isopropyl alcohol of at least 90% purity. The easiest way is to use cassette head cleaner/demagnetizer tape sold by most audio departments and record stores.

Clean diskette drives periodically. The use of the fluid on the swab is not recommended. Unless you know how to work on these machines, use a head cleaner diskette. Follow the instructions on the package carefully to avoid damage to the unit.

Have disk drives tested periodically. Diskette drives do go out of alignment. This causes them to miss important data because they are looking in the wrong place. They also write data in the wrong place so it cannot be found by other drives. This must be adjusted by a trained technician. Have it done before it becomes a problem. Once or twice a year provides good insurance against lost data. Cassette drives may also go out of alignment but the problem is less severe. If you are experiencing loss of data, first clean the heads. If that fails, have the head alignment checked by a technician.

Care for the monitor. Aside from cleaning the face of the tube, there are several things that need to be done. Make sure that the display is easy to read. Most monitors, or television sets, have contrast and brightness controls. Adjust them for maximum viewing comfort. Eventually, however, the adjustment may no longer work. A competent TV repair shop may be able to make internal adjustments, giving the set a longer, useful life. If all else fails, perhaps a new picture tube may solve the problem.

There are focusing controls built into all of these sets, but they are inside. If your screen looks fuzzy, have a technician check these controls.

A NOTE ABOUT CABLES AND CONNECTORS

Most computers have at least one cable connection to peripherals such as cassette drive, diskette drive, or printer. These cables rarely cause problems; however, if a problem arises, it may be a major problem. The following are some notes on the care of cables and their connectors.

Never bend a cable sharply. This is especially true of the broad, flat ribbon cables. Each conductor in the cable is quite tiny. A sharp bend can cause the wire inside to break. Such a break, while completely unseen by you, will be noticed immediately by the computer. Finding such a break can be a time-consuming and maddening job. An ounce of prevention.....

Be careful plugging and unplugging. The connectors on the ribbon cables are often broad, flat plugs called card-edge connectors. When plugging or unplugging these cables, always push or pull the connector, NEVER the cable. If the connector is stubborn, apply firm pressure on the ends. You may rock the connector from end to end, but NEVER wiggle it the short way. Such movement can break the circuit board, and that requires the complete replacement of the board, an expensive item. These plugs are found on the disk drives of the Apple II and TRS-80 model I, and on the printers attached to most computers. The PET uses card-edge connectors attached to round cables for the disk drives and/or printer, and on the cassette drives.

Check for cord placement. Be careful where the cords run. Power and peripheral cords that cross the floor in traffic areas are a hazard. Students and teachers can trip and serious personal injury can result. Or a computer may be knocked over and damaged.

Appendix F

Glossary of Computing Terms

abacus	A device used to aid in arithmetical calculations. It consists of a counting frame with parallel wires or rods on which sliding balls or beads are moved.
access	To get; to access a record is to get a record.
acoustical coupler	A modem that connects a computer terminal to a telephone handset or receiver and changes computer signals to signals that can be sent over a phone line. (see modem)
address	A number for a location in the computer's memory. When programming a computer, an address is given to each piece of information and to each instruction. In the problem 2 + 3 = the 2 and 3 have addresses at which they are located. The answer will have its own address and so will the instruction that tells the computer to add these numbers.
Aiken, Howard	The man who developed the Mark I mechanical computer. It was also known as ASCC (Automatic Sequence Controlled Calculator); one of many pioneers of modern computer design.
algorithm	A procedure for solving a problem or making a decision.
application programmer	A programmer who learns how programs will be used (that is, the application) so that the programs he or she writes will work well.
analog computer	A computer that works with a continuous scale or measure as opposed to the digital computer, which works with a scale or measure with distinct values. A watch that uses hands to tell time is an analog watch. A digital watch shows the time with numbers only.
Atanasoff, John V.	An early pioneer in computing who, with Clifford Berry, experimented with the first electronic computer (the Atanasoff-Berry computer), which used vacuum tubes.

Babbage, Charles	A British mathematician who designed a calculating machine called the Analytical Engine.
batch processing	Gathering information and entering it into a computer all at once as opposed to entering it continuously. Order forms and bills, are often batch processed.
BASIC	(Beginner's All Purpose Symbolic Instructional Code) A programming language for beginning programmers.
benchmark run	A test run of a program used to judge its performance. It may be measured against a different program, or against its own performance on another type of computer.
Berry, Clifford	The man who, with Atanasoff, experimented with the first electronic computer.
binary number system	A number system using 0 and 1 to represent all numbers. For example: 2 in the decimal system is 10 in the binary system; 3 is 11 in the binary system.
bit	A short way of saying binary digit, the smallest unit of information a computer processes. A bit is either a 1 or a 0, one of which means current is flowing in the circuit and the other means it is not.
branch	To select one of several possible sets of instructions in the execution of a computer program. For example, the computer may follow one set of directions if a number is greater than 5 and another set of directions if a number is less than 5.
bubble memory	A solid state device that uses tiny bubbles or magnetized areas to store information; a type of secondary storage device.
bug	A programming term for an error or mistake.
byte	A number of bits processed as a unit and usually shorter than a computer word. A byte is often 8 bits long.

card punch	A device that punches holes in a card to represent data.
card reader	An input device that senses hole patterns in a punched card and changes them into computer data.
census	An official counting of the population. This occurs every 10 years in the United States
central processing unit	(CPU) The part of the computer that interprets instructions and carries them out; the core of the computer.
character	A letter, digit, or other symbol that is used to represent data. All symbols that appear on a computer keyboard are characters. A space is also a character.
chip	A common name for a small bit of silicon upon which an integrated circuit is built. A chip is about one-half the size of your smallest fingernail.
COBOL	(Common Business Oriented Language). A standardized language developed for business use.
coding	A way of shortening data so it will take fewer characters or memory locations. This is done by using a single number, the coded value, to represent a range of original values. For instance, the ages 1 year through 5 years may all be represented by the number 1.
compiler	A program that changes a computer program written in another language, such as FORTRAN or COBOL, into machine languge.
(electronic) computer	A machine that stores and processes information using electrical and magnetic signals.
computer science	The study of theories of computation. Computer scientists develop languages and ways of putting data into the computer, among other things.

conditional structures	Instructions that test the values of variables and change the execution of a program accordingly. For example, a program may say if the result (or branch) is greater than 5, go to location 16; if it is less than 5 go to location 17.
controller	A device that moves information between a disk or tape and the main memory so the CPU can use it.
control structure	A set of instructions and a statement that tells how many times or when the set of instructions is to be executed.
customer engineer	A technician who makes sure a computer always works properly. Computer owners usually hire the services of a customer engineer from the company that sold them the computer.
data	The information that is put into a computer to be processed by a program or that results from the execution of a program.
data flowchart	A flowchart that shows the path of data in the solving of a problem, and which also notes the phases of processing.
data processing	Taking information and changing it into other useful forms. Data processing may include calculation and information storage, among other things, to produce the desired results.
decimals	A system of numbers using a base of 10 so that all numbers are represented by combinations of the numerals 0 through 9.
device	A machine made for a certain purpose, such as secondary storage device.
digit	Any one of the 10 Arabic numerals 0 through 9.
direct access files	(Also called random access files) Files from which records may be taken in any order as opposed to sequential files, in which a record may be found only by going through each preceding record in order.

digital computer	A computer that works with a scale or measure with distinct values as opposed to the analog computer, which works with a continuous scale or measure. A digital watch shows the time with numbers only. An analog watch is one with hands.
disk	A flat, circular plate (usually the size of a long playing phonograph record) with a magnetizable surface layer upon which data can be stored by magnetic recording; a secondary storge device.
diskette	A secondary storage device (about the size and shape of a 45 RPM record) with a magnetic surface for recording.
disk drive	A device that plays and records on disks or diskettes.
direct connect modem	A modem that connects directly to the phone line and changes computer signals to signals that can be sent over a telephone line.
Ecker, J. Presper	The man who developed the ENIAC, a first generation computer, with John W. Mauchly.
execute	To perform or carry out, as to execute (or run) a computer program.
execution or run-time	An error that can only be found by executing or running the program.
feedback	The use of part of the output of a machine or system as input to make it self-correcting.
file	A set of related records stored together and treated as a unit. For example, a file may contain all the payroll records for a department of a company.
field	A certain area of a record used for a particular piece of data such as age or name, for example.
field service engineer	A technician who will travel to a company to fix its computer. Companies that sell computers employ field service engineers to serve their customers.

field service personnel	The customer support people employed by companies that sell computers. They teach customers how to use new programs and they fix the computer system if it breaks down.
field test	The testing of a computer program on a very small scale so errors or problems may be found and changed before the program is used widely.
first generation computer	A computer that uses vacuum tubes, like the Atanasoff-Berry computer, the ENIAC, and the UNIVAC.
finite	Having a definite beginning and an end.
flowchart	A graphic representation (or picture) of an algorithm using symbols to show the steps.
FORTRAN	(FORmula TRANslation) A computer language used mostly by scientists.
graphic digitizer	A graphic input device that changes lines or pictures on paper into coordinates that are automatically fed into the computer.
graphic tablet	A kind of graphic digitizer made up of a pen or stylus used to trace on an electronic board or "tablet."
graphic terminal	A terminal that shows graphic information on a tv-like screen. It may show bar graphs or maps, for example.
hardware	The physical equipment of a computer (as opposed to software).
hardware designer	Electronics engineer who designs the physical equipment of computers.
hard copy	A printed copy of machine output in a readable form, such as listings or reports.
Hollerith, Herman	The man who developed a punched card system for the census. This punched card system was revolutionary. It enabled people to store and retrieve large amounts of information.

Hollerith card	A punched card designed by Herman Hollerith to store information; used today by computers.
information	Organized data used in a computer program.
instruction	In a programming language, a command that names one operation and identifies its operands.
information error	A mistake in a program caused by a mistake in the data given the computer.
input	Information given to the computer, or program results that are recycled as information to be further processed; to input information is to put it into the computer.
input/output devices	Devices that move data into or out of the main memory of a computer.
integrated circuit	A small chip of silicon that performs the same functions as many transistors, capacitors, and resistors on a circuit board.
Jacquard, Joseph Marie	The man who used punched cards to control looms. This was the first automated machine.
Josephson circuit	A new type of circuit, which may increase the speed of computers as much as 100 times.
jump	A change in the usual order in which instructions are followed; a branch. For example, a computer may follow one set of instructions if a number is greater than 5 and another set of instructions if the number is less than 5.
key	The portion of a record used to determine where a record goes in a file.
kilobyte	1024 bytes of information. The memory capacity of a computer is often indicated in terms of how many kilobytes it can store.
language	A set of words and other symbols used to give instructions to a computer. Each computer language has its own structure.
Leibnitz, Gottfried Wilhelm von	The man who created an early multiplying and dividing machine. This was a great improvement on Pascal's adder and dramatically increased the speed at which calculations could be made.

light pen	A graphic input device that is light sensitive. When the light pen touches the display (or tv) screen, the computer can sense the position of the pen.
line printer	A device that prints a line of characters as a unit, as opposed to printing one letter or character at a time as a typewriter does.
loop	A set of instructions that may be executed repeatedly while a certain condition exists. For instance, as long as a certain number in the program remains under 5, the program keeps repeating the same steps.
Lovelace, Ada Augusta	Countess Lovelace, the daughter of the poet Lord Byron, saved Babbage's ideas by writing about them in everyday language.
machine language	The only language understood directly by a computer. Machine languages are composed of binary numbers. Each type of computer has its own machine language.
main memory	(Sometimes called core memory). The internal storage device of the computer.
Mauchly, John W.	The man who developed ENIAC, a first generation computer, with Eckert.
mechanical plotter	A graphic output device that uses pen and ink to draw lines on paper.
megabyte	1,048,576 bytes of information, often used to note the size of memory.
menu	A menu in a computer program is similar to the table of contents in a book; it tells what the program includes.
microcomputer	A computer that has a microprocessor as its central processing unit.
microprocessor	A tiny silicon chip that is the central processing unit of a computer.
minicomputer	Computers that range in size from the size of a desk to the size of a phone booth. Minicomputers are often used with a variety of input and output devices to enable them to serve very specific purposes. They are very flexible.

modem	Short for modulator/demodulator; changes computer signals (digital data) into tones that can be carried on telephone lines. It also changes computer signals that come over the phone as tones back into digital data to be entered in a computer.
multiplier	The numeral that tells how many times a step is repeated.
Neumann, John von	The man who proposed that computers use the binary number system and stored programs. The use of these two ideas changed the direction of computer design and greatly speeded its development.
online	A terminal connected to a computer at which a person may enter data continuously. Some online terminals are connected to a computer via a telephone and are only linked in when there is information to be processed.
operand	A piece of data from the computer's memory to be used by a machine language instruction.
operating systems	A program that controls the execution of computer programs.
operation	One of the basic functions a computer can perform such as addition, subtraction, and movement of data.
operation code	The portion of a machine language that tells the computer what operation to perform as opposed to where to get the data.
operations manager	One who supervises computer operators, makes sure the computer has enough paper, and so on, and generally manages the work flow.
operations personnel	Personnel who are directly responsible for making sure the computer is usable and operating properly.
operational	Able to be used; an operational algorithm is one that tells exactly what to do at each step and is therefore usable.
output	The information, or results generated by a computer during the running of a program.

Pascal, Blaise	The man who created an early adding and subtracting machine. This machine was the first step in the development of computers.
personal computer	(Also called microcomputer). A tabletop computer that has a microprocessor as its central processing unit.
peripheral devices	External devices that can extend a computer's capabilities. These include input/output and secondary storage devices.
plotter	An output device that presents information in a graphic form by drawing mechanically with pen and ink on paper.
power	The speed and storage capacity of a computer. The greater the speed and storage capacity of a computer, the greater its power.
process	A systematic order of operation followed to produce a certain result. When a computer adds or otherwise performs operations on numbers, it is processing the numbers.
program	(n) The instructions that tell the computer what to do; (v) to write the instructions that tell the computer what to do.
program counter	A special register in the CPU that holds the address of the next instruction to be executed.
quantification	The changing of data to numbers or measurable quantities.
record	A collection of related data treated as a unit, such as the payroll information for one person.
register	Very fast memory locations found in the central processing unit.
repetition	Repeating programs steps a number of times as a "control" that makes programs shorter or easier to write.
secondary storage devices	Supplementary, external data storage or memory, such as magnetic tape or disks.

second generation computers	Computers that use transistors, capacitors, and resistors instead of vacuum tubes or integrated circuits. The IBM 7090 is a second generation computer.
sequential file	A file of information that can only be read record by record, starting from the beginning of the file. To reach a certain record, the computer must first go through all of the other records in front of that one.
simulation	In computer technology, the development of a model for the purpose of testing a theory.
software	computer programs
software houses	Companies that specialize in creating and selling computer programs.
solid state device	An electrical part that works by the flow of electrons in solids; for example, a transistor or a bubble memory.
stored programs	Instructions that tell the computer how to process data. These are stored like data in the computer.
subroutine	A self-contained set of instructions used by a program or by another subroutine.
supervisory program	A computer program that controls the execution of other computer programs and controls the flow of work in a data processing system. Part of the operating system.
syntax error	An error caused by a mistake in the way a command or information is given to the computer. Each programming language has its own structure which must be followed. If the structure is not followed, the programmer has made a syntax error.
system	A collection of people, machines, and methods organized to meet certain goals.
systems analysis	Procedures for the systematic study of the structure of a system to improve it. A study of the relationship of the purpose of parts to the system's goals. Also, a procedure for designing new systems.

systems analyst	One who studies a system and suggests ways to improve that system.
systems programmer	One who creates operating systems and languages for computers.
tabulate	To arrange in a table; to condense and list.
tape drive	A device similar to an audio tape recorder that records and plays magnetic tape; used to record and store computer programs and data.
technical representative	A person who knows about computers and about a certain field, such as manufacturing or medicine, and can help a customer select the best computer for his or her purpose.
terminal	A device, usually equipped with a keyboard and some kind of display, that can send information to and receive information from a computer.
time sharing	Use of a computer by more than one person at the same time.
transistor	A small semi-conducting device that performs the functions of a vacuum tube in a circuit. Among other things, it can serve as a switch or a memory device.
third generation computers	Computers that use integrated circuits or silicon chips. Most minicomputers and all microcomputers are third generation computers.
unambiguous	Straightforward, can be interpreted in only one way.
vacuum tube	A sealed tube (used in electronics) which contains metallic electrodes but almost no air or gas.
variable	A part of a program that is changeable or may have a number of values.
video terminal	A terminal with a television-like screen for showing information.
word	A unit of data storage noted in terms of either bits or bytes; varies in length from one model of computer to another.
word processor	An office computer used to manipulate words or written material rather than numbers.

Appendix G

Glosario de terminos usados en computadoras

Inglés	Español	Definición
Abacus	Ábaco	Un aparato para contar y calcular. Consiste en un armazón con cuerdas o varillas paralelas por las cuales se deslizan bolitas o cuentas.
Access	Acceso	Obtener información de la computadora.
Acoustical coupler	Modulador acústico	Un aparato al cual se le conecta el auricular de un teléfono y cambia las señales de la computadora a señales que se pueden enviar a través de los cables telefónicos.
Address	Dirección (señal)	Número para indicar localización en la memoria de una computadora. Cuando se programa en una computadora, una dirección es dada a cada trozo de información y a cada instrucción. En el problema 2 + 3 =, el número 2 y el número 3 tienen direcciones en la computadora mediante las cuales se pueden localizar éstos. La contestación tiene su propia dirección al igual que la instrucción y a su vez le indica a la computadora que sume estos números.
Aiken, Howard	Howard Aiken	El inventor de la computadora mecánica Mark I, la cual se le conoce también como ASCC (Calculadora Automática de Secuencia Controlada). Es uno de los pioneros del diseño de la computadora moderna.
Algorithm	Algoritmo	Procedimiento utilizado para resolver un problema o llegar a una decisión. Se emplea para calcular sumas, diferencias, productos, cocientes, raíces cuadradas, etc.
Application programmer	Programador	Persona que sabe y conoce de programas y de su uso adecuado de tal manera que al diseñar los programas funcionen de acuerdo al propósito destinado.

Inglés	Español	Definición
Analog computer	Computadora analógica	Una computadora que trabaja con una medida o escala continua en contraposición a una digital, la cual trabaja con medidas o escalas de valores precisos. Un reloj de agujas es un reloj analógico. Un reloj digital marca la hora con números.
Atanasoff, John V.	John V. Atanasoff	Uno de los precursores de las computadoras electrónicas quién en compañía de Clifford Berry, experimentó con la primera de estas máquinas (el Atanasoff - Berry) que usaba válvulas electrónicas.
Babbage, Charles	Charles Babbage	Matemático inglés que inventó una máquina computadora llamada "el motor analítico".
BASIC	BÁSICO	El lenguaje básico de la programación de computadoras para principiantes el cual es usado por la mayoría de las microcomputadoras.
Batch processing	Procesar información acumulada	Acumular información y entrarla del todo en una computadora, contrario a entrar información de una manera continua. Órdenes así como recibos de cuentas son ejemplos de información acumulada para procesar.
Benchmark run	Prueba de utilidad de un programa	Correr una prueba de un programa a usarse para medir su utilidad. Dicho programa puede juzgarse en relación a otro programa o el propio rendimiento que este mismo programa haya tenido en otra computadora.
Berry, Clifford	Clifford Berry	Científico quién junto a Atanasoff experimentó con la primera computadora electrónica.
Binary Number System	Sistema Numérico Binario	El sistema numérico de base 2 que usa el 0 y el 1 para representar a todos los números. Por ejemplo, el número 2 en el sistema decimal es 10 en el sistema binario, así como el 3 es 11 en el sistema binario.

Inglés	Español	Definición
Bit		Una manera abreviada de decir cifra binaria. Es la unidad de información más pequeña que una computadora puede procesar. Un "bit" es 1 ó 0, el primero significando el paso de corriente y el segundo la ausencia de ella.
Branch	Ramas o dirección	Una de las posibles direcciones o ramas en las que una computadora puede ejecutar su programa. Por ejemplo, si el número es mayor que el 5, la computadora puede seguir un grupo de instrucciones diferentes, así como otro grupo de instrucciones si el número es menor que el 5.
Bubble memory	Memoria en burbujas o en áreas magnetizadas	Dispositivo que usa pequeñas burbujas o áreas magnetizadas para almacenar datos. Un dispositivo de memoria secundario.
Bug	Error de lógica	Vocablo usado por programadores para designar un error.
Byte		El número de "bits" procesados por la computadora como una unidad. Frecuentemente es 8 "bits".
Card punch	Perforador de tarjeta	Aparato que perfora huecos en una tarjeta para así representar datos.
Card reader	Lector de las perforaciones	Dispositivo sensible a los huecos en una tarjeta perforada que cambia la información contenida en ellos a data de la computadora.
Census	Censo	La cuenta oficial de la población. En los E.U. se efectúa cada 10 años.
Central processing unit (CPU)	Unidad central de procesamiento (UCP)	La parte de la computadora que interpreta instrucciones y las ejecuta. El cerebro de la máquina computadora.
Character	Tipo, signo, símbolo	Una letra, digíto u otro símbolo que representa datos. Todo lo que aparece en el tablero de una computadora son tipos, signos o símbolos. El espacio se considera también un símbolo.

Inglés	Español	Definición
Chip	Ficha electrónica	Nombre genérico que se asigna a una pequeña pieza de silicio en la cual se construye un circuito integrado. Su tamaño no excede la mitad de una uña del dedo meñique.
COBOL (Common Business Oriented Language)	COBOL	Lenguaje desarrollado para uso comercial que ha sido uniformado (normalizado).
Coding	Codificar	Una manera para reducir data de manera que la menor cantidad de símbolos o memoria sean utilizados. Esto se realiza utilizando solamente un número, el valor clave, para representar una serie de valores originales. Por ejemplo, las edades entre 1 año y 5 años pueden ser representados por el número 1.
Compiler	Compilador	Un programa usado para traducir otro programa escrito en un lenguaje dado tales como FORTRAN o COBOL, a lenguaje de máquina.
(Electronic) computer	Computadora electrónica	Máquina que almacena y procesa datos por medio de señales eléctricas o magnéticas.
Computer science	Ciencia de computadoras	El estudio de teorías y métodos usados en programar las computadoras electrónicas modernas. Los científicos de computadoras han desarrollado entre otras cosas lenguajes, análisis numéricos, y formas para procesar datos en una computadora.
Conditional structures	Estructuras condicionales	Instrucciones que examinan los valores de variables y de acuerdo con ellos alteran el orden con que un programa es ejecutado. Por ejemplo, un programa puede decidir ir al lugar 16 si un resultado es mayor que 5 o ir al 17 si el resultado es menor que 5.
Controller	Interventor o controlador	Un artefacto que transfiere información entre un disco o cinta y la memoria central de tal manera que la unidad central (C.P.U. - Central

Inglés	Español	Definición
Controller (cont'd)	Interventor (cont.)	Processing Unit) pueda usarla.
Control structure	Estructura de control	Un conjunto de instrucciones y una exposición que indica cuando o cuántas veces un grupo de instrucciones van a ser ejecutadas.
Customer engineer	Técnico de computadoras	Un técnico encargado del funcionamiento correcto de la computadora. Normalmente los propietarios de las computadoras usan los servicios de técnicos empleados por la compañía manufacturera.
Data	Datos	La información que se introduce en la computadora para ser procesada por un programa; el resultado de la ejecución de un programa.
Data flowchart	Diagrama de datos	Un diagrama que muestra el curso que los datos siguen en la solución de un problema y que también indica las etapas del procesamiento de la información.
Data processing	Procesamiento de datos	Usar información y cambiarla a otras formas más útiles. El procesamiento de datos puede incluir entre otras cosas cálculos matemáticos y almecenamiento de datos, de manera que se produzcan los resultados buscados.
Decimals	Decimales	Un sistema numérico de base 10. Todos los números se representan con combinaciones de los numerales del 0 al 9.
Device	Artefacto, aparato, máquina	Un artefacto o máquina hecho con algún propósito como por ejemplo, dispositivo de almacenaje secundario.
Digit	Dígito	Cualquier numeral arábigo (0 - 9).
Direct access files	Archivo de acceso directo	También llamados ("random access files") acceso al azar. Son aquellos archivos cuyos expedientes se encuentran en cualquier orden; en contraposición a los archivos en secuencia, cuyos expedientes se pueden encontrar solamente pasando a través de todos

Inglés	Español	Definición
		los expedientes que preceden al deseado.
Digital computer	Computadora digital	Una computadora que funciona con una escala o medida de valores precisos en contraposición a una computadora analóga que usa una escala continua.
Disk	Disco	Disco circular plano con una superficie magnetizada en la cual se almacenan los datos. Un artefacto de almacenaje secundario.
Diskette	Disco pequeño	Un aparato de almacenaje secundario el cual es aproximadamente del tamaño y forma de un disco de 45 rpm.
Disk drive	Activador de información	Aparato que lee y graba en disco de diferentes tamaños.
Direct connect modem	Modulador de conección directa	Un modulador que se conecta directamente a la línea del teléfono y cambia señales de la computadora a señales que pueden ser enviadas a través de la línea telefónica.
Ecker, J. Presper	J. Presper Ecker	El científico que junto a John W. Mauchly inventó las primeras computadoras llamada "ENIAC".
Execute	Ejecución	Ejecutar, llevar a cabo o realizar un programa de computadoras.
Execution or Run-time	Detectar un error en un programa	Es un error que puede ser encontrado solamente si se ejecuta o se corre el programa.
Feedback	Corrección directa de un programa	El uso de parte de la potencia de salida o resultados de un programa o sistema como potencia de entrada para producir un proceso de auto-corrección.
File	Archivo	Un conjunto de expedientes almacenados considerados como una unidad. Por ejemplo, el conjunto de todas las nóminas de una compañía.
Field	Campo específico de cierta data	Cierta área de un expediente usada para almacenar un dato en particular

Inglés	Español	Definición
		como por ejemplo, la edad o el nombre.
Field service engineer	Técnico de servicio al cliente	El técnico que se dedica a visitar las las compañías que han comprado computadoras para arreglarlas cuando es necesario. Este técnico rinde servicios a los clientes de la compañía a la cual él representa.
Field service personnel	Personal de adiestramiento	Empleados de compañías manufactureras que adiestran a los clientes en el uso de los nuevos programas y que arreglan defectos ocurridos a las computaras.
Field test	Proceso de prueba	Proceso por el cual se examina o ejecuta un programa de computadora a menor escala. El propósito es el de encontrar y corregir posibles errores antes de usar el programa a mayor escala.
First generation computer	Primeras computadoras computadora	La computadora que usa válvulas electrónicas, como por ejemplo, la de Atanasoff - Berry o la de ENIAC y la de UNIVAC.
Finite	Finito	Finito, que tiene principio y fin.
Flowchart	Diagrama	Representación gráfica o un diagrama de un algoritmo con símbolos que muestra el progreso de un proceso.
FORTRAN	Término de computadora que significa fórmula y traducción	(FORmula-TRANslation) Lenguaje especial de computadoras usado por matemáticos, ingenieros, físicos y científicos en general.
Graphic digitizer	Instrumento de valores numéricos gráficos	Un aparato de entrada que cambia líneas o dibujos en un papel en coordenadas que son suministradas automáticamente a la computadora.
Graphic tablet	Tablero para gráficas	Un aparato o artefacto como los anteriores que usa una pluma especial para delinear en un tablero electrónico.
Graphic terminal	Terminal gráfico	Artefacto semejante a un receptor de televisión. Muestra información grá-

Inglés	Español	Definición
		fica como histógramos o mapas.
Hardware	Equipo y accesorios de una computadora	Piezas y accesorios que constituyen la computadora propiamente dicha. Es lo opuesto de "software" que se relaciona con los programas de computadoras.
Hardware designer	Diseñadores de computadoras	Técnicos electrónicos que diseñan las computadoras electrónicas.
Hard copy	Copia legible	Una copia legible impresa de datos de una computadora. Ejemplo: listas, tablas o informes.
Hollerith, Herman	Herman Hollerith	El que inventó la tarjeta perforada para ser usada en el censo. Este sistema fue de índice revolucionario ya que permitía almacenar y encontrar enormes cantidades de datos e información.
Hollerith card	Tarjeta perforada Hollerith	Una tarjeta perforada como la mencionada anteriormente. Llamada así en honor de su inventor.
Information	Información	Organización de datos usados en un programa de computadoras.
Information error	Error de información	Error en un programa de computadoras causado por datos erróneos.
Input	Entrada de datos (en computadora)	Información suministrada a la computadora o el resultado de un programa cuya información vuelve a ser procesada; esto es volver a introducir la información dentro de la computadora.
Input-output	Aparatos de entrada/	Son artefactos que tienen la capacidad de entrar o sacar información de la memoria principal de una computadora.
Instruction	Instrucción	Una orden en un lenguaje dado que obliga a la computadora a efectuar una operación identificando al mismo tiempo sus parámetros.
Integrated circuit	Circuito integrado	Una pequeña ficha de silicio que realiza las mismas funciones que los transistores, condensadores y resistencias en un circuito.

Inglés	Español	Definición
Jackard, Joseph Marie	Joseph Marie Jackard	El científico que usó por primera vez tarjetas perforadas para controlar información borrosa. Fue la primera máquina automática.
Josephson circuit	Circuito de Josephson	Un tipo nuevo de circuito que puede aumentar por más de 100 veces la velocidad de una computadora.
Jump	Cambio de orden en instrucciones	Es un cambio en el orden en el cual las instrucciones son ejecutadas por una computadora. Por ejemplo, la computadora puede seguir un conjunto determinado de instrucciones si un número es mayor que 5 y ejecutar un conjunto diferente de instrucciones si el número es menor que 5.
Key	Clave	Parte de un expediente que determina donde se colocará la información en el archivo.
Kilobyte		La capacidad de memoria de una computadora es por lo general expresada en términos de la cantidad de "kilobytes" que puede almacenar, o sea, 1024 "bytes" de información.
Language	Lenguaje	Es un grupo de palabras y otros símbolos que se usan para dar instrucciones a una computadora. Cada lenguaje de una computadora tiene su propia sintaxis.
Leibnitz, Gottfried Wilhelm Von	Gottfried Wilhelm Von Leibnitz	El que inventó la primera calculadora para multiplicar y dividir. Esto representó un gran paso de avance en la máquina de sumar de Pascal lo cual ayudó a aumentar la rapidez para hacer cálculos matemáticos.
Light pen	Pluma lumínica	Un aparato gráfico con potencia de entrada que es sensitivo a la luz. Cuando esta pluma toca la pantalla de televisión, la computadora percibe su posición.
Line printer	Impresor de línea	Un aparato que imprime una línea de símbolos como una unidad, en contraposición a una máquina de escribir que imprime símbolo a símbolo.

Inglés	Español	Definición
Loop		Un conjunto de instrucciones que pueden ser ejecutados repetidamente bajo ciertas condiciones existentes. Por ejemplo, siempre y cuando que cierto número en un programa permanezca siendo un número menor de 5 el programa continuará repitiendo los mismos pasos.
Lovelace, Ada Augusta	Ada Augusta Lovelace	La condensa Lovelace, hija del poeta Lord Buyron, preservó las ideas de Babbage escribiéndolas en un lenguaje simple.
Machine language	Lenguaje de la computadora	El único lenguaje que puede ser entendido por una computadora. Los lenguajes de estas computadoras están compuestos por números binarios. Cada tipo de computadora posee su propio lenguaje.
Main memory	Memoria principal	La memoria interna de una computadora electrónica. Muchas veces llamada el núcleo de la memoria.
Mauchly, John W.	John W. Mauchly	El diseñador de ENIAC, quién junto a Eckert desarrolló la primera generación de computadoras.
Mechanical plotter	Gráfico mecánico	Un aparato gráfico de salida que usa una pluma y tinta para trazar líneas en un papel.
Megabyte		El "megabyte" es utilizado para determinar el tamaño de la memoria de una computadora. Un "megabyte" equivale a 1,048,576 "bytes" de información.
Menu	Índice de programas de computadora	Similar al índice de un libro. Nos indica lo que el programa incluye.
Micro-computer	Microcomputadora	Una computadora que tiene un "microprocesor" como su unidad central para procesar información.
Micro-processor	Microprocesor	Una ficha pequeña de silicio la cual es la unidad central para procesar información en una computadora.

Inglés	Español	Definición
Minicomputer	Minicomputadora	Computadoras que varían en tamaño desde una talla semejante a una mesa hasta el de una cabina telefónica. Se usan con una gran variedad de dispositivos de entrada y salida los cuáles les permiten ser muy flexibles.
Modem	Modulador	Abreviatura de modulador/demodulador. Cambia señales digitales (de la máquina) a pulsos que pueden ser usados en líneas telefónicas. También efectúa este mismo proceso a la inversa.
Multiplier	Multiplicador	El numeral que indica cuántas veces se repite un paso.
Neumann, John von	John von Neumann	Fue el que propuso el uso del sistema binario y el almacenaje de programas en las computadoras. Estas dos ideas cambiaron la dirección en el diseño de las computadoras y como resultado, se aceleró grandemente el desarrollo de éstas.
Online	Terminal de computadora	Un terminal conectado a una computadora por el cual una persona le suministra datos a la máquina continuamente. Algunos terminales están conectados por medio de un teléfono y solo entran en acción cuando hay información para ser procesada por la computadora.
Operand	Operando	Parte de los datos de una información en la memoria de la computadora para ser usados formar parte de una instrucción en un lenguaje de máquina.
Operating systems	Sistemas operacionales	Programa que controla la ejecución de otros programas en una computadora electrónica.
Operation	Operación	Una de las funciones básicas que puede efectuar una computadora, tales como suma, resta y comparaciones o movimiento de datos.

Inglés	Español	Definición
Operation code	Clave de operación	Parte del lenguaje de una computadora que indica a la máquina cual operación se va a efectuar, en contraposición de donde se obtienen los datos.
Operations manager	Gerente de operaciones	Es el supervisor del personal que opera la computadora de tal manera que el trabajo realizado por este personal sea adecuado. Es el que se encarga de supervisar que la computadora esté equipada con el material necesario para dar servicio eficientemente.
Operations personnel	Personal de operaciones	Empleados cuya responsabilidad radica en que la computadora rinda buen servicio y funcione adecuadamente.
Operational	Operacional	Estar disponible para usarse; un algoritmo operacional que expresa claramente qué hacer a cada paso y que por lo tanto puede usarse.
Output	Información de salida de una computadora	La información o resultados producidos por una computadora durante la ejecución de un programa.
Pascal, Blaise	Blaise Pascal	Filósofo y matemático quién inventó la primera máquina de sumar y restar. Este invento marcó el primer paso en la evolución de las computadoras.
Personal computer	Computadora personal	También llamada una microcomputadora. Una computadora en miniatura que tiene un microprocesor como la unidad central para procesamiento de información.
Peripheral devices	Dispositivo periférico	Dispositivos exteriores que aumentan la capacidad para funcionar de una computadora. Entre éstas se incluyen aparatos de entrada y salida de información de las computadoras, así como también aparatos secundarios de almacenaje.
Plotter	Dispositivo gráfico	Dispositivo de salida que convierte datos en gráficas mediante el uso de una pluma que dibuja mecánicamente en el papel.

Inglés	Español	Definición
Power	Potencia	Velocidad y capacidad de memoria de una computadora. A mayor velocidad y capacidad de almacenaje de una computadora, mayor es su potencia.
Process	Proceso	Un orden sistemático de operación que se sigue para llegar a un posible resultado. Cuando una computadora suma o lleva a cabo operaciones con números, está procesando los números.
Program	Programa	El conjunto de instrucciones que le indican a una computadora como hacer un trabajo determinado; escribir las instrucciones que le indican a la computadora lo que va a hacer.
Program counter	Contador	Un contador especial en la unidad central de la computadora (CPU) que contiene la dirección de la próxima instrucción que se va a ejecutar.
Quantification	Cuantificación	Transformación de datos a números u otras cantidades que pueden ser medidas.
Record	Registro	Un conjunto de datos considerados como una unidad, como por ejemplo, la información concerniente a la nómina de una persona.
Register	Registro	Direcciones rápidas en la memoria de una computadora las cuales se encuentran en la Unidad Central de Procesar Información (CPU).
Repetition	Repetición	Repetir los pasos de un programa un sinnúmero de veces como un control para poder hacer los programas más cortos o más fáciles de escribir.
Secondary storage devices	Aparatos de memoria o almacenaje secundario	Aparatos suplementarios para almacenar datos o memoria tales, como cintas magnéticas o discos.
Second generation computers	Computadoras de la segunda generación	Computadoras que usan transistores, condensadores y resistencia en lugar de válvulas o circuitos integrados. Ejemplo: IBM 7090

Inglés	Español	Definición
Sequential files	Archivos en secuencia	Archivo de información que puede leerse solamente en secuencia, o sea desde el principio a fin. Para conseguir una información determinada la computadora tiene que pasar primeramente toda la información que precede a la información deseada.
Simulation	Simulación (modelo de prueba)	La creación de un modelo con el propósito de probar una teoría con la ayuda de una computadora electrónica.
Software	Programas y materiales	Todos aquellos artículos, materiales y programas que se usan en una computadora.
Software houses	Compañías de programas de producción	Compañías que se especializan en la producción y venta de programas.
Solid state device	Aparato de estado sólido	Dispositivo sólido que trabaja mediante el flujo de electrones en ciertos materiales sólidos como por ejemplo los transistores o las válvulas de memoria.
Stored programs	Programas almacenados	Las instrucciones que indican a la computadora como procesar los datos. Éstas se almacenan como datos en la memoria de la computadora.
Supervisory program	Programa de control	Un programa que controla la ejecución de otros programas y el movimiento de trabajo en un sistema de procesar datos o información.
Syntax error	Error de sintaxis	Error ocasionado por la forma incorrecta en que una orden o información ha sido dada a la computadora. Cada lenguaje tiene su estructura la cual hay que seguir para evitar incurrir en un error de sintaxis.
System	Sistema	Conjunto de personas, máquinas y datos organizados para satisfacer ciertos objetivos.
System analysis	Análisis sistemático	Procedimientos para el estudio sistemático de la estructura de un sistema con el fin de mejorarlo. Estudio de

Inglés	Español	Definición
		la relación entre objetivos y partes del sistema. Procedimientos para nuevos sistemas.
System analyst	Analista de sistema	El que estudia o analiza un sistema y sugiere maneras para mejorarlo.
Systems programmer	Programador de sistema	El que desarrolla sistemas y lenguajes para usarse en las computadoras electrónicas.
Tabulate	Tabulación	Ordenar en forma de tabla; condensar y enumerar.
Tape drive	Grabador magnético	Aparato similar a una grabadora magnética. Usado para grabar y almacenar datos y programas.
Technical representative	Representante técnico	Persona que tiene conocimientos sobre computadoras y otros campos tales como la medicina o la manufactura. También puede ayudar a una persona a seleccionar la computadora que mejor se adapte a sus necesidades.
Terminal	Terminal	Aparato que normalmente contiene un tablero y pantalla usado para recibir o transmitir información de la computadora.
Time sharing	Uso simultáneo	Uso de una computadora por más de una persona al mismo tiempo.
Transistor	Transistor	Un pequeño dispositivo semi-conductor que realiza las mismas funciones que una válvula en un circuito. Entre otras cosas puede hacer de conmutador o dispositivo de memoria.
Third generation computers	Computadoras de tercera generación	Computadoras que usan circuitos integrados o fichas de silicio. Muchas de las minicomputadoras y todas las microcomputadoras pertenecen a esta categoría.
Unambiguous	Inequívoco	Claro de expresión, sin ambigüedades. Que puede ser interpretado de una sola manera.
Vacuum tube	Válvulas electrónicas	Tubo sellado utilizado en electrónica que contiene electrodos metálicos y carece de aire o gas en su interior.

Inglés	Español	Definición
Variable	Variable	Parte de un programa que puede ser alterado o que puede tener diferentes valores.
Video terminal	Terminal de video	Un terminal con pantalla de televisión para mostrar información.
Word	Información en palabras	Una unidad para el almacenamiento de datos en términos de "bits o bytes." Varía en longitud de acuerdo con el modelo de la máquina.
Word processor	Fototipografía	Una computadora usada en oficinas con el objeto de escribir, producir y almacenar información por un procedimiento fotomecánico.

Appendix H

Directory of Selected Software Production Companies

Acorn Software Products, Inc.
634 N. Carolina Ave., S.E.
Washington, D.C. 20003
202-544-4259

Activity Resources Inc.
P.O. Box 4875
Hayward, CA 94540
415-782-1300

Addison-Wesley Publishing Company
2725 Sand Hill Rd.
Menlo Park, CA 94025
415-854-0300

Advanced Operating Systems
450 St. John Rd.
Michigan City, IN 46360
219-879-4693

Adventure International
507 East St.
Box 3435
Longwood, FL 32750
800-327-7172

Apple-Cations
21650 W. Eleven Mile Rd.
Suite 103
Southfield, MI 48076
313-354-2559

Apple Computer Inc.
10260 Bandley Drive
Cupertino, CA 94017
408-996-1010

Artworx Software Co.
150 N. Main St.
Fairport, NY 14450
800-828-6573

Atari, Inc.
1272 Borregas Ave.
Sunnyvale, CA 94086
408-745-5069

Atari Program Exchange
P.O. Box 427
155 Moffett Dr., B-1
Sunnyvale, CA 94086

Avant-Garde Creations
P.O. Box 30161
Eugene, OR 97403
503-345-3043

Bargainbyte
P.O. Box 23195
Harahan, LA 70183

Basics and Beyond, Inc.
Box 10
Amawalk, NY 10501
914-962-2355

Bell & Howell
7100 N. McCormick Rd.
Chicago, IL 60645
312-262-1600

Bluebird's Computer Software
2267 23rd St.
Wyandotte, MI 48192
313-285-4455

Borg-Warner Educational Systems
600 W. University Dr.
Arlington Heights, IL 60004
800-323-7577

Brain Box
601 W. 26th St.
New York, NY 10003
212-989-3573

California Software
P.O. Box 275
El Cerrito, CA 94530
415-527-8717

Class 1 Systems
17909 Maple Street
Lansing, IL 60438
312-474-4664

Commodore Computer
681 Moore Rd.
King of Prussia, PA 19406
215-337-7100

Compress
P.O. Box 102
Wentworth, NH 03282
603-764-5831

Compumax, Inc.
P.O. Box 1139
Palo Alto, CA 94301
415-321-2881

Computer-Advanced Ideas, Inc.
1442A Walnut St.
Suite 341
Berkeley, CA 94709
415-526-9100

Computer Aided & Managed Instruction
P.O. Box 2030
Goleta, CA 93118

Computer Consulting
6723 E. 66th Pl.
Tulsa, OK 74133
914-492-9036

Computer Curriculum Corp.
P.O. Box 10080
Palo Alto, CA 94303
415-494-8450

Computer-ED
1 Everett Rd.
Carmel, NY 10512
914-225-1158

Computer Information Exchange
P.O. Box 159
San Luis Rey, CA 92068
714-757-4849

Computer Learning Connection
One Boston Place
Boston, MA 02108
617-723-4433

Computer's Voice
2370 Ella Dr., Dept. 122
Flint, MI 28504
313-238-5585

Computronics
50 N. Pascack Rd.
Spring Valley, NY 10977
914-425-1535

Conduit
P.O. Box 388
Iowa City, IA 52240
319-353-5789

Control Data Corporation
8100 34th Ave. South
P.O. Box 0
Minneapolis, MN 55440
612-853-4541

Cook's Computer Company
1905 Baily Dr.
Marshalltown, IA 50158

Creative Computing
39 E. Hanover Ave.
Morris Plains, NJ 07950
800-631-8112

Creative Software
201 San Antonio Circle 270
Mountain View, CA 94040
415-948-9595

Creative Thamaturgy
P.O. Box 107 Forest Park Branch
Dayton, OH 45405

Curriculum Applications
P.O. Box 264
Arlington, MA 01174

Cybertronics International
999 Mt. Kemble Ave.
Morristown, NJ 07960
201-766-7681

Dr. Daley's Software
Water St.
Darby, MT 59829
406-821-3924

Developmental Learning Materials
One DLM Park
Allen, TX 75002
214-248-6300

Digiac Corporation
175 Engineers Rd.
Smithtown, NY 11787
516-273-8600

Dresen Associates
P.O. Box 246
Dresen, ME 04342
207-737-4466

Dynacomp
1427 Monroe Ave.
Rochester, NY 14618
716-442-8960

Education Programs
Disney Electronics
6153 Fairmont Ave.
San Diego, CA 92120
714-281-0285

Educational Activities
1937 Grand Ave.
P.O. Box 87
Baldwin, NY 11510
516-223-4666

Educational Computing Systems
106 Fairbanks Rd.
Oak Ridge, TN 37830
615-483-4915

Educational Courseware
3 Nappa Lane
Lees Port, CT 06880
203-227-1438

Educational Micro Systems
P.O. Box 471
Chester, NJ 07930

Educational Services Management C
P.O. Box 12599
Research Triangle Park, NC 27709
919-781-1500

Educational Software Midwest
414 Rosemere Lane
Maquoketa, IA 52060
319-652-1334

Educational Software Professionals, Ltd.
38437 Grand River
Farmington Hills, MI 48018
313-447-4470

Edu-Soft Educational Software
4639 Spruce St.
Philadelphia, PA 19139
215-747-1284

Edutek Corporation
P.O. Box 11354
Palo Alto, CA 94036
415-325-9965

Edu-Ware Services, Inc.
P.O. Box 22222
Agoura, CA 91301
213-346-6783

Electronic Courseware Systems, Inc.
P.O. Box 2374 Station A
Champaign, IL 61820
217-359-7099

Entelek
Ward-Whidden House
The Hill
P.O. Box 1303
Portsmouth, NH 03801
603-436-0439

Etronix
14803 N.E. 40th St.
Redmond, WA 98052
206-881-0857

J.L. Hammett Company, Inc.
Hammett Place
P.O. Box 545
Braintree, MA 02184
617-848-1000

Hartley Software
P.O. Box 431
Dimondale, MI 48821
616-942-8987

Hayden
50 Essex St.
Rochelle Park, NJ 07662
201-843-0550

High Technology Software Products, Inc.
P.O. Box 14665
8001 N. Classen Blvd.
Oklahoma City, OK 73113
405-840-9900

Holt-Rhinehart, Winston
383 Madison Ave.
New York, NY 10017
212-872-2206

Houghton Mifflin Company
One Beacon St.
Boston, MA 02107
617-725-5000

Information Unlimited Software
281 Arlington Ave.
Berkeley, CA 94707
415-525-9452

Instant Software
Peterborough, NH 03458
603-924-7296

Instructional Development Systems
29 Virginia Beach Blvd.
Virginia Beach, VA 23452
804-340-1977

International Institute of Applied Technology, Inc.
20010 Century Blvd.
Germantown, MD 20767
301-428-9010

International Software Marketing, Ltd.
120 E. Washington St.
Syracuse, NY 13202
315-474-3400

Interpretive Education
2306 Winters Drive
Kalamazoo, MI 49002
616-345-8681

J&S Software
140 Reid Ave.
Port Washington, NY 11050
516-944-9304

Krell Software
21 Milbrook Drive
Stonybrook, NY 11790
516-751-5139

Learning Tools, Inc.
4 Washburn Place
Brookline, MA 02146
617-566-7585

Level IV Products, Inc.
32461 School Craft
Livonia, MI 48150
313-525-6200

Liberty Software Company
635 Independence Ave., SE
Washington, DC 20003
202-544-6674

Lightning Software
P.O. Box 11725
Palo Alto, CA 94306
415-856-1855

Little Genius
34-38 83rd St.
Jackson Heights, NY

Logo Computer Systems
989 Avenue of the Americas
New York, NY 10018
212-564-6020

L & S Computerware
1589 Fraser Drive
Sunnyvale, CA 94087
408-738-3416

McGraw-Hill
1221 Avenue of the Americas
New York, NY 10020
212-997-6194

Charles Mann & Associates
55722 Santa Fe Trail
Yucca Valley, CA 92284
714-365-9718

Math Software
1233 Blackthorn
Deerfield, IL 60015

Mathware/Math City
4040 Palos Verdes Dr. N.
Rolling Hills Estates, CA 90274
213-541-3377

MECC
2520 Broadway Drive
St. Paul, MN 55113
612-376-1122

Mentor Software
Box 791
Anoka, MN 553303

Mercer Systems, Inc.
87 Scooter Lane
Hicksville, NY 11801

Merry Bee Communications
815 Crest Drive
Omaha, NE 68046
402-592-3479

MicroGnome
5843 Montgomery Rd.
Elkridge, MD 21227
301-796-2456

Microlab
2310 Skokie Valley Rd.
Highland Park, IL. 60035
312-433-7550

Micro Learningware
P.O. Box 2134
N. Mankato, NM 56001
507-625-2205

Microphys Programs
2048 Ford St.
Brooklyn, New York 11229
212-646-0140

Micro Power & Light Company
12820 Hillcrest Rd. #224
Dallas, TX 75230
214-239-6620

Microsoft Consumer Products
10700 Northup Way
Bellevue, WA 98004
206-828-8080

Milliken Publishing Company
Computer Department
1100 Research Blvd.
St. Louis, MO 63132
314-991-4220

Milton Bradley
443 Shaker Rd.
East Longmeadow, MA 01028
413-525-6411

Monument Computer Service
Village Data Center
P.O. Box 603
Joshua Tree, CA 92252
800-854-0561

Muse Software
347 N. Charles St.
Baltimore, MD 21201
310-659-7212

National Software Marketing
4701 McKinley St.
Hollywood, FL 33021
305-625-6062

Opportunities For Learning, Inc.
8950 Lurline Ave.
Chatsworth, CA 91311
213-341-2535

Optimized Systems Software
10379 Lansdale Ave.
Cupertino, CA 95014
408-446-3099

Prentice Hall
Sylvan Ave.
Englewood Cliffs, NJ 07632
201-592-2649

Prescription Learning
1301 S. Wabash Ave.
Chicago, IL 60605
312-922-0579

Program Design, Inc.
11 Idar Court
Greenwich, CT 06830
203-661-8799

Programs For Learning
P.O. Box 954
New Milford, CT 06776
203-355-3452

Program Research and Software Corp.
257 Central Park West
New York, NY 10024
212-787-1526

Quality Education Design
P.O. Box 12486
Portland, OR 97212
503-287-8137

Radio Shack
1800 One Tandy Center
Fort Worth, TX 76102
817-390-3832

Random House
2970 Brandywine Rd.
Suite 201
Atlanta, GA 30341

Rell
1125 Stanford Ave.
Redondo Beach, CA 90278

Resource Software International
140 Sylvan Ave.
Englewood Cliffs, NJ 07632
201-947-6104

Right On Programs
P.O. Box 977
Huntington, NY 11743

Science Research Associates (SRA)
155 North Wacker Drive
Chicago, IL 60606
800-621-0665

Scott, Foresman & Company
1900 East Lake Ave.
Glenview, IL 60025
312-729-3000

Serendipity Systems
225 Elmira Rd.
Ithaca, NY 14850
607-277-4889

Simutek Computer Products, Inc.
4877 E. Speedway
Tucson, AZ 85712
602-323-9391

Sliwa Enterprises, Inc.
P.O. Box 400
Big Flats, NY 14814
607-562-3287

Software Exchange
6 South St.
Milford, NJ 03055

SoftWare House, Inc.
695 East 10th North
Logan, UT 84321
800-453-2708

Solartek
P.O. Box 298
Guilderland, NY 12084

Southern Micro-Systems for Educators
P.O. Box 1981
Burlington, NC 27215
919-226-7610

SouthWest EdPsyche Services
P.O. Box 1870
Phoenix, AZ 85001

Sterling Swift Publishing Company
1600 Fortview Rd.
Austin, TX 78704

Storybooks of the Future
527 41ST Ave.
San Francisco, CA 94121
415-386-5184

TARA
P.O. Box 118
Selden, NY 11784
516-331-2537

Teacher's PET Software
1517 Holly St.
Berkeley, CA 94703

Terrapin, Inc.
678 Massachusetts Ave.
Cambridge, MA 02139
617-494-8816

Texas Instruments, Inc.
Personal Computer Division
P.O. Box 53
Lubbock, TX 79408
800-858-4565

T.H.E.S.I.S.
P.O. Box 147
Garden City, MI 48135
313-585-4722

3 R Software
P.O. Box 3115
Jamaica, NY 11431

Time Share Corporation
Hanover, NH 03755
603-448-3838

TYC Software
40 Stuyvesant Manor
Geneseo, New York 14454
716-243-3005

Tycom Associates
63 Velma Avenue
Pittsfield, MA 01201

Unicom
297 Elmwood Avenue
Providence, RI 02907
401-467-5600

Universal Systems for Education
2120 Academy Circle, Suite E
Colorado Springs, CO 80909
303-574-4575

Versa Computing, Inc.
3541 Old Conejo Rd., Suite 104
Newbury park, CA 91320
805-498-1956

Window, Inc.
469 Pleasant St.
Watertown, MA 02172
617-923-9147

XPS, Inc.
323 York Rd.
Carlisle, PA 17013
717-243-5373

Appendix J

A Brief History of Calculating and Computing

This appendix highlights important advances in the evolution of calculating devices and computers. This background information will be useful in teaching the three history lessons: 2A, 2B, and 2C.

Early Aids to Calculation

Early people used their fingers and toes to aid in counting, but this was limiting and allowed people only to count up to 20.

With more possessions to keep track of and count (for example, shepherd's flock), people began using stones, pebbles, lines drawn on the ground, and tally sticks (sticks with notched markings). However, stones and pebbles were heavy and easily lost, lines on the ground disappeared with the winds, and wood was needed for kindling.

About 5,000 years ago (3,000 B.C.), people invented the abacus (Chinese) or soroban (Japanese). This device has two sets of beads strung on rods. The beads represent various numbers -- for example, the beads on the first rod may represent the ones unit; those on the second rod, the tens units, etc. Different numbers are represented by sliding the beads to different positions on the rods. This device was probably the first to use a place-value system.

Number Systems

Eventually, people invented number systems for counting. The Romans used numerals such as I, V, X, L, C and D. In the thirteenth century, the Hindu Arabic numeral system was invented. It used the digits 0 through 9 and had a place-value system. This system is still used today.

Calculating Devices

In 1642, Blaise Pascal, whose father was a French tax collector, invented the first mechanical calculating machine using digits. By using wheels that turned to different numbers, the machine could either add or subtract, but not both. Nevertheless, it helped his father with his tax accounts. About 50 were built. Twenty years later (1662), Samuel Morland invented a mechanical calculator that was used to compute taxes in France. This machine was an improvement over Pascal's. It could both add (wheels turned clockwise) and subtract (wheels turned counterclockwise).

By the end of the seventeenth century (1694), a German, Gottfried von Leibniz, invented a calculator called the "reckoning machine" which could do all four mathematical operations. It was very expensive but was much faster than Pascal's calculator. Only two were built.

About a hundred years later (1805), Joseph Jacquard, a French weaver, invented small cards made of stiff paper. Each card was punched with holes. The pattern of the holes became the pattern on the cloth that he wove using a loom. By using these cards, exactly the same pattern could be woven over and over again.

By 1842, after working for 20 years, Charles Babbage, an Englishman, designed an automatic calculator which he called the analytical engine. this machine was a giant leap forward. It had the potential to compute, store, or remember numbers for later use and print answers. Numbers were to be "fed"

BEST COPY AVAILABLE

into the analytical engine using punched cards similar to those invented by Jacquard. The engine was too costly and technology not advanced enough to build Babbage's design. Ada Byron Lovelace worked with Babbage, corrected some errors in his work, and wrote about his ideas in everyday language. Her writings became the basis for constructing a model of the analytical engine many years later.

United States law requires that the Census Bureau count the number of people in the U.S. every decade. The 1880 census took seven and a half years to complete and the bureau was concerned that work on the 1890 census would still be in progress when time came for the 1900 census. In 1887, Herman Hollerith, a Census Bureau worker, invented a punched card and tabulating machine that enabled the 1890 census to be completed in two and a half years. Hollerith went on to form his own company, which later became part of IBM (International Business Machines). Punched cards are still used today to feed information into computers, as one way of inputting.

Evolution of the Computer

In the last 40 years, there have been great advances in the computing field. In 1944, the MARK I computer was built. It was based on Babbage's ideas and used electromechanical switches to do calculations. It could add three eight-digit numbers in a second. In 1946, ENIAC (Electronic Numerical Integrator and Calculator) became the first all-electronic computer. Instead of using moving wheels, gears, and levers, ENIAC used vacuum tubes (gas-filled glass tubes) for power. It weighed 60,000 lbs (30 tons) and could do 5,000 additions a second. The ENIAC operated until 1955 when it was disassembled and part of it sent to the Smithsonian Institution in Washington, D.C.

EDVAC, designed by John von Neumann in 1949, was the first electronic digital computer to use the binary (two-digit) number system (0 = light switch off; 1 = light switch on) and capable of storing, or remembering, number directions. The first computer designed for commercial use, UNIVAC I, was delivered to the U.S. Census Bureau in 1951.

By the late 1950s, "first generation" computers using vacuum tubes could do 40,000 addition problems per second. A "second generation" (1958-1964) of computers evolved as vacuum tubes were replaced by transistors and solid state amplifiers. Computers became smaller, less expensive, and faster (200,000 additions per second). Transistors required less power, gave off less heat, and lasted longer than vacuum tubes. A "third generation" of computers began in 1964 when IBM introduced the integrated circuit which reduced the electronic components of the computer to about the size of 1/2 inch ceramic tile. Computers could now do 1,250,000 additions per second. In the late 1970s, hundreds of thousands of integrated circuits were combined into one unit called a chip. A chip is made of silicon, a common material found in the earth's surface, and is about 1½ times the size of your tiny fingernail. Computers can now do 10,000,000 additions per second. The recent development of the Josephson circuit will make computers even more powerful, perhaps 100 times faster than computers using integrated circuits.

Computers Today

Early computers were often as large as a room. Today there are computers that fit into a pocket. Compared to ENIAC, computers today are 2,000 times faster, thousands of times more reliable, consume the power of a light bulb rather than that of a locomotive, occupy 1/30,000 the volume and cost 1/10,000 as much. Big industries and the government use large, costly computers called mainframes or maxicomputers.

Some businesses and schools use computers called minicomputers. Most classroom computers are even smaller and cheaper than the others. They are called microcomputers.

Types of Computers Developed Since 1944 and Their Capacities

DATE	COMPUTER	TYPE	SPEED (ADD/SEC)
1944	MARK I	ELECTROMECHANICAL	3
1946	ENIAC	ELECTRONIC (VACUUM TUBE)	5,000
1964	IBM 7090	ELECTRONIC (TRANSISTOR)	100,000
1970	IBM 360	ELECTRONIC (INTEGRATED CIRCUIT)	1,000,000
1980	CRAY I	ELECTRONIC (SILICON CHIP)	10,000,000
?	?	ELECTRONIC (JOSEPHSON)	100,000,000

Evolution of Computer Languages

As computers became more powerful and cheaper, computer languages were developed to help people communicate with computers. In the late 1950s, FORTRAN was developed to help solve scientific problems and COBOL to help solve business problems. By 1964 a new simple, all-purpose computer language called BASIC was developed. More recent languages include LOGO, PILOT, and PASCAL.

PROFESSIONAL COMPUTER ORGANIZATIONS

Association for Computing Machinery (ACM)
1133 Avenue of the Americas
New York, NY 10036
212-265-6300

Association for Educational Communication and Technology (ACET)
1126 16th Street, N.W.
Washington, DC 20036
202-833-4180

Association for Educational Data Systems (AEDS)
1201 16th Street, N.W.
Washington, DC 20036
202-833-4100

Computer Using Educators (CUE)
Independence High School
1775 Educational Park Drive
San Jose, CA 95113

International Council for Computers in Education (ICCE)
Department of Computer and Information Science
University of Oregon
Eugene, OR 97403
503-686-4414

National Council of Teachers of Mathematics (NCTM)
1906 Association Drive
Reston, VA 22901
703-620-9840

Appendix K

Bibliography

Periodicals

BYTE
Box 590, Martinsville NJ 08836

Classroom Computer News
Intentional Educations, Inc., 341 Auburn Street, Watertown, MA 10172

Computing Teacher
Department of Computer and Information Science, University of Oregon, Eugene, OR 97403

Compute
P.O. Box 5406, Greensboro, NC 27403

Creative Computing
P.O. Box 789-M, Morristown, NJ 07960

Electronic Learning
902 Sylvan Ave., Englewood Cliffs, NJ 07632

Infoworld
375 Cochituate Rd., Framingham, MA 01701

Instructor
P.O. Box 6099, Duluth, MN 55806

Personal Computing
4 Disk Drive, Box 13916, Philadelphia, PA 19101

Popular Computing
P.O. Box 307, Martinsville, NJ 08836

****The following magazines are machine specific. They are written for a particular computer.*****

APPLE

Apple Educator's Newsletter
9525 Lucerne Street, Ventura, CA 93004

Apple Orchard
International Apple Corps, P.O. Box 976, Daly City, CA 95017

Call-Apple
304 Main Ave. S., Suite 300, Renton, WA 98055

In-Cider
80 Pine Street, Peterborough, NH 03458

Nibble
Box 325, Lincoln, MA 01773

Peelings II
945 Brook Circle, Las Cruces, NM 88001

Softalk
11201 Magnolia Blvd., North Hollywood, CA 91601

ATARI

A.N.A.L.O.G.
400/800 P.O. Box 23, Worcester, MA 01603

Antic
297 Missouri Street, San Francisco, CA 94107

Atari Connection
Atari Home Computer Division, 60 E. Plumeria Ave., San Jose, CA 95134

Purser's Magazine
P.O. Box 466, El Dorado, CA 95623

COMMODORE PET (Vic-20 and Com. 64)

Commodore Magazine
Commodore Systems Division, 681 Moore Rd., King of Prussia, PA 19406

Compute
(specific for Apple, Atari, and PET)
P.O. Box 5406, Greensboro, NC 27403

IBM

PC
1528 Irving Street, San Francisco, CA 94122

TRS-80 (Radio Shack)

Color Computer Magazine
Highland Hill, Camden ME 04843

TRS-80 Microcomputer News
700 One Tandy Center, Ft. Worth, TX 76102

80 Microcomputing
P.O. Box 981, Farmingdale, NY 11737

80-US
3838 S. Warner Street, Tacoma, WA 98409

Softside
(specific for Apple, Atari, and TRS-80)
P.O. Box 68, Milford, NH 03055

TEXAS INSTRUMENTS

99'er Magazine
P.O. Box 5537, Eugene, OR 97405

Personal Computer Newsletter
P.O. Box 53, Lubbock, TX 79408

Books

**Textbooks -- Classroom use -- (T) and (E) elem.; (U) upper)
Reference, resource -- (R)

Abelson, Harold. *LOGO for the Apple.* Peterborough, N.H.: McGraw-Hill, 1982. (R,T,E,U)

Albrecht, Mary Jo. *My Computer Likes Me.* Menlo Park, Calif.: Dymax, 1972. (T,E)

Albrecht, Robert; Finkel, L.; and Brown, G. *BASIC for Home Computers.* New York: John Wiley and Sons, 1978. (R)

Ball, Marion. *What Is a Computer?* Boston: Houghton Mifflin Co., 1972. (T,E)

Ball, Marion, and Charp, S. *Be a Computer Literate.* Morristown, N.J.: Creative Computing Press, 1977. (T,E)

Billings, Karen, and Moursund, D. *Are You Computer Literate?* Beaverton, Ore.: Dilithium Press, 1979. (R)

Brown, Jerald R. *Instant BASIC.* Beaverton, Ore.: Dilithium Press, 1982. (T,U,E)

Carlson, Edward H. *Kids and the Apple Computer.* (other editions for Atari and PET) Reston, Va.: Reston Publishing, 1982. (T,E,U)

Deken, Joseph. *The Electronic Cottage.* New York: William Morrow and Co., 1981. (R)

D'Ignazio, Fred. *Katie and the Computer.* Morristown, N.J.: Creative Computing Press, 1979. (T,E)

Doerr, Christine. *Microcomputers and the 3 R's: A Guide for Teachers.* Rochelle Park, N.J.: Hayden Book Co., 1980. (R)

Dwyer, Thomas, and Critchfield, M. *BASIC and the Personal Computer.* Reading, Mass.: Addison-Wesley, 1978. (R)

Dwyer, Thomas, and Critchfield, M. *A Bit of BASIC.* Reading, Mass.: Addison-Wesley, 1980. (T,R,U)

Evans, Christopher. *The Micro Millennium.* New York: Viking Press, 1979. (R)

Grady, Tim, and Gawronski, Jane. *Computers in Curriculum and Instruction.* Alexandria, Va.: ASCD Publications, 1983. (R)

Jones, Aubrey B. *I Speak BASIC to My Apple.* (other editions for TRS-80, PET, VIC, Atari) Rochelle Park, N.J.: Hayden Book Co., 1982. (T,E,U)

Kosel, Marge, and Carlstrom, G. Elementary, My Dear Computer. St. Paul, Minn.: Minnesota Educational Computing Consortium, 1978. (R)

Larsen, Sally. Computers for Kids. (available in three seperate editions for Apple, Atari, and TRS-80) Morristown, N.J.: Creative Computing Press, 1981. (T,E)

Luehrmann, Arthur, and Peckham, Herbert. Computer Literacy: A Hands-on Approach. (Apple and TRS-80 editions) New York: McGraw-Hill, 1983. (T,U)

McWilliams, Peter A. The Personal Computer Book. Los Angeles, Calif.: Prelude Press, 1982. (R)

Moursund, David. Teacher's Guide to Computers in the Elementary School. La Grande, Ore.: Computing Center, Eastern Oregon State University, 1980. (R)

Papert, Seymour. Mindstorms: Children, Computers and Powerful Ideas. New York: Basic Books, 1980. (R)

Presley, Bruce. A Guide to Programming in Applesoft. (other editions for IBM, TRS-80, DEC) New York: Van Nostrand Reinhold, 1982. (T,E)

Rice, Jean. My Friend the Computer. Minneapolis, Minn.: T.S. Dennison, 1976. (T,E)

Rice, Jean, and O'Connor, S. Computers Are Fun. Minneapolis, Minn.: T.S. Dennison, 1981. (T,E,U)

Richardson, Kay. One Computer, Thirty Kids. Indianapolis, Ind.: Meka Publishing Co., 1980. (R)

Richman, Ellen. Spotlight on Computer Literacy. New York: Random House, 1982. (R,T,E,U)

Taylor, Robert. The Computer in the School: Tutor, Tool, Tutee. New York: Academic Press, 1977. (R)

Thornburg, David. Picture This! PILOT Turtle Geometry. Reading, Mass.: Addison-Wesley, 1982. (T,R,E,U)

Zausmer, Russell N. Centering On Microcomputers: An Introductory Guide. New York: New York City Teacher Centers Consortium, 1982. (R)

**** In addition to the above books, your best source of information for your particular computer is your Owner's Manual and any publication that the manufacturer publishes for your machine. Also, almost daily, there are articles in newspapers and magazines about computer use in the schools.

Evaluation Questionnaire

Computer Literacy (Intermediate and Secondary Grades)

This is an experimental bulletin in computer education. It is a pioneering effort to standardize computer instruction on a citywide basis. Your suggestions for improvement are most welcome.

1. How many classes in your school pursued this course of study? __________

2. What were the grade levels of the students taking this course?

   ____

3. Is the format of the bulletin useful? ______________________

4. Is the bulletin well organized? ______________________

5. Are the goals of the computer literacy course clearly stated?

   ____

6. Are the activities appropriate for the intended grade levels?

   ____

7. Is there sufficient emphasis on enduring computer concepts, such as good documentation, structured programming, modular design, problem-solving skills, and so forth? ______________________

   ____

8. Did the learning activities enable students to achieve the performance objectives? ______________________

   ____

9. Was there a significant variety of activities in the learning strategies?

   ____

   ____

10. What changes (if any) would you make in the learning activities presented in the bulletin? ______________________

    ____

11. What additions would you make to the scope of the course content?

    ____

    ____

12. What topics in the course of study would you delete or change?

13. Was the interest level of the materials high among the students?

14. What changes in sequence would you recommend?

15. What difficulty (if any) did you encounter in implementing this course?

16. What do you consider to be the overall strengths of the bulletin? (If any)

17. What do you consider to be the overall weaknesses of the bulletin? (If any)

18. What changes would you suggest to improve the quality of the bulletin? (If any)

19. Please make any additional comments you feel are appropriate.

Return to: Mr. Marc Licht, Director
Computer and Information Sciences Unit
131 Livingston Street - Room 409
Brooklyn, New York 11201